

DocStore

by aMedia Corp.



DocStore

aMedia사의 DocStore는 고객이 문서를 업로드하고 관리하는 인터넷 서비스입니다.

- 어디서든지 콘텐츠를 업로드하고 액세스 – 브라우저와 모바일 앱
- PDF, 오피스 문서, 이미지, 비디오 – 텍스트 내용 추출, 인덱싱 및 검색 가능

DocStore

두 가지 계정 유형

- 무료
 - 5GB 스토리지
 - 파일명, 생성/수정 날짜, 태그로 검색
 - 내용 검색 불가
 - 모든 콘텐츠는 소유자가 안전하게 다운로드 가능



두 가지 계정 유형

- **프리미엄**

- 20GB 스토리지
- 모든 문서에 대해 내용 검색 지원(이미지에 대해서는 OCR 적용)
- 모든 콘텐츠는 소유자가 안전하게 다운로드 가능





두 가지 주요 구성 요소

DocStore

#1: 웹 애플리케이션

- 사용자가 모든 문서를 업로드, 관리, 조회할 수 있도록 허용
- 모든 콘텐츠는 S3로 직접 업로드

DocStore

#1: 웹 애플리케이션

- 퍼블릭 DocStore API를 통해 기능 제공
- 확장성 지원 필수
- 고가용성 지원 필수



DocStore

#1: 웹 애플리케이션

- 글로벌 지원을 위해 복수 지역에 배치
- 어떤 지역에서도 사용자 로그인 가능
 - 항상 '홈' 지역으로 리디렉션(즉, 계정이 초기에 생성된 지역으로)

DocStore

#1: 웹 애플리케이션

- DocStore API를 포함하는 웹 애플리케이션과 모바일 애플리케이션을 위한 아키텍처를 그룹 활동을 통해 설계

DocStore 요구 사항

계정 유형

- **무료** – 5GB 용량, 마스터 문서 인덱싱, 원본 파일 다운로드
- **프리미엄** – 20GB 용량, 내용 검색 + OCR

웹 애플리케이션/API

- 사용자 인증
- 인스턴스 외부에 세션 상태 저장
- API 계층을 통해 기능 제공
- 고가용성

콘텐츠

- S3 내에 저장
- 효율적인 업로드/다운로드
- 정적 자산은 하나의 S3 버킷에 저장되지만 글로벌하게 전송될 수 있어야 함

배치

- 오레곤(us-west-2)과 시드니(ap-southeast-2)
- 최소한의 정보만 중복 저장(즉, 업로드된 콘텐츠는 중복 저장하지 않음)





샘플 아키텍처

2개 지역에 애플리케이션 배치: us-west-2(오레곤) 및
ap-southeast-2(시드니).

DocStore



각 ELB는 Route 53에 등록되는 지역별 CNAME을 가집니다.

DocStore



Route 53의 LBR(지연 시간 기반 라우팅) 기능은 고객을 가장 가까운 단말점으로 리디렉션합니다.

DocStore



Amazon Route
53

amdocstore.com
www.amdocstore.com

 **amazon**
web services™

오레곤이 더 가까우면? 오레곤으로 리디렉션합니다. 호주 퍼스에서 접속하면? 시드니의 ELB로 리디렉션합니다.

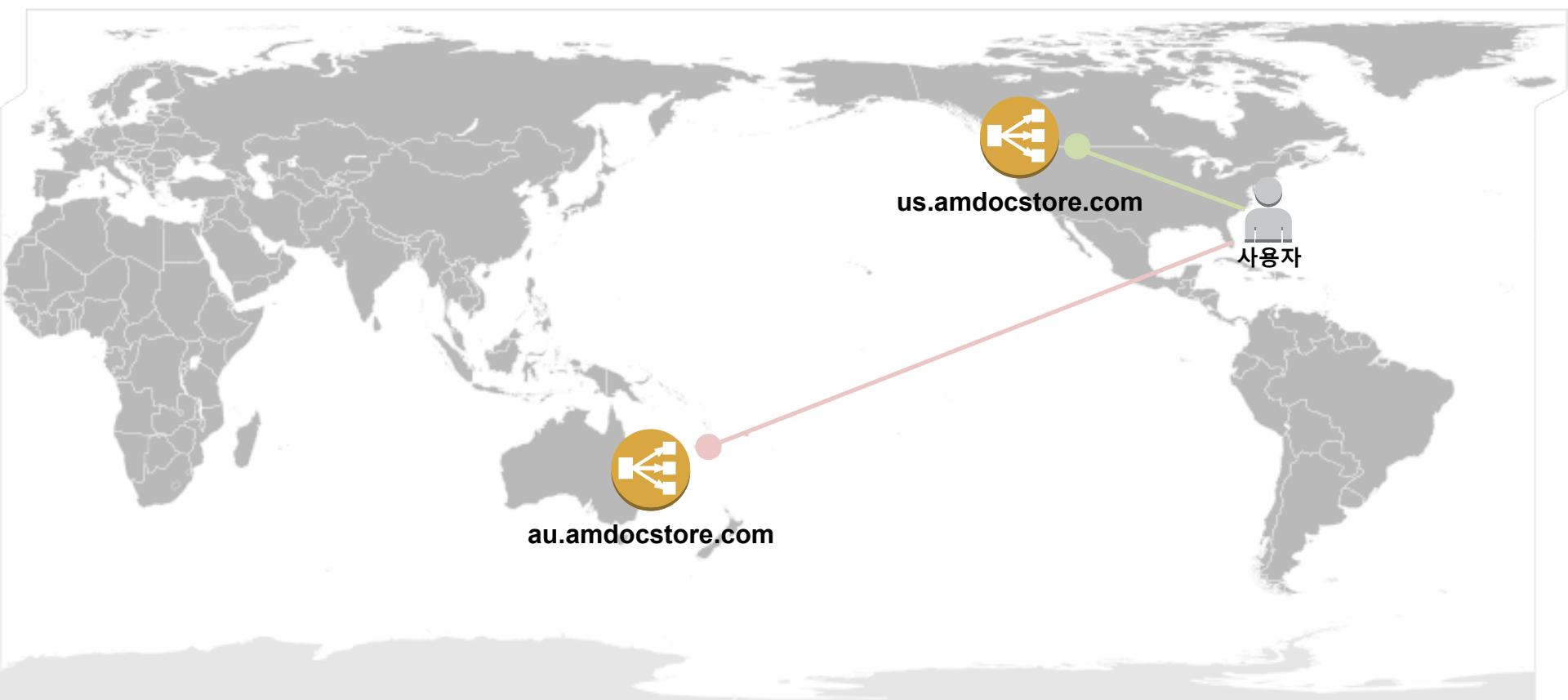
DocStore



Amazon Route
53

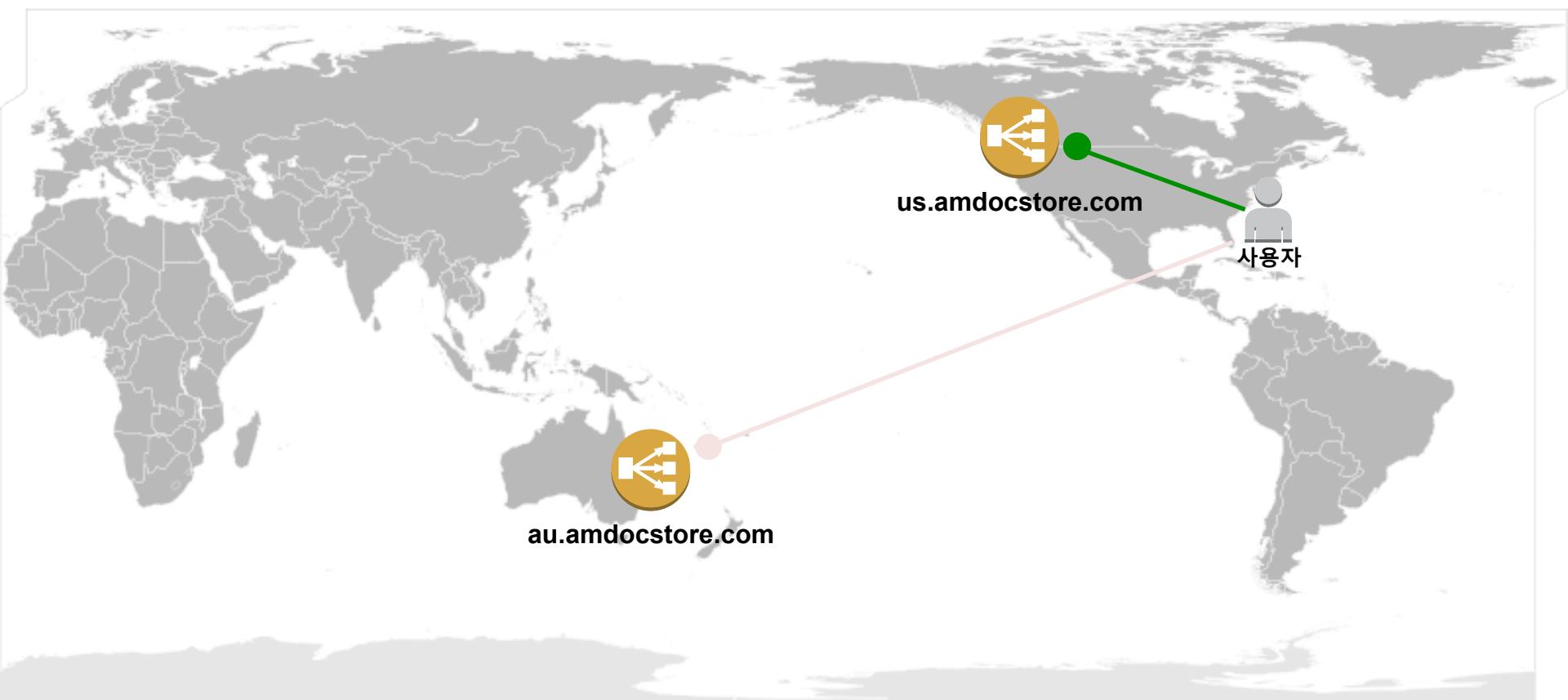
오레곤이 더 가까우면? 오레곤으로 리디렉션합니다.

DocStore



오레곤이 더 가까우면? 오레곤으로 리디렉션합니다.

DocStore



하나의 지역에 집중해 보겠습니다.

DocStore

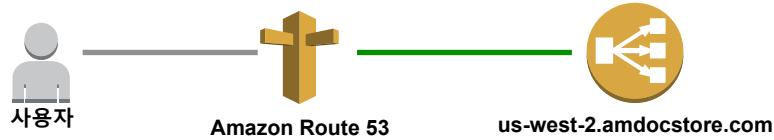


us.amdocstore.com



Route 53의 LBR은 가장 가까운 Elastic Load Balancer로 사용자를 리디렉션합니다(이 경우 us-west-2의 ELB).

DocStore



DocStore는 모든 정적 웹 자산(CSS, 이미지, JavaScript 등)을
S3에 저장합니다.

DocStore



amdocstore-global-assets
.s3.amazonaws.com

amazon
webservices™

이 버킷은 us-west-2에 존재하지만 글로벌하게 요청되는 정적 자산을 위한 정규 소스(canonical source)입니다.

DocStore

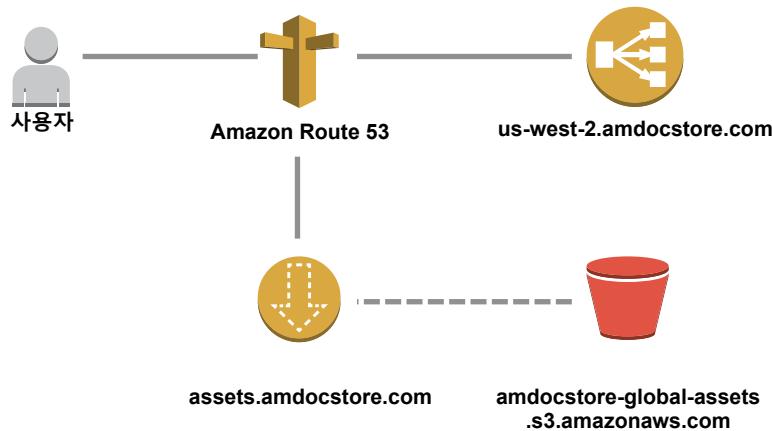


amdocstore-global-assets
.s3.amazonaws.com

amazon
webservices™

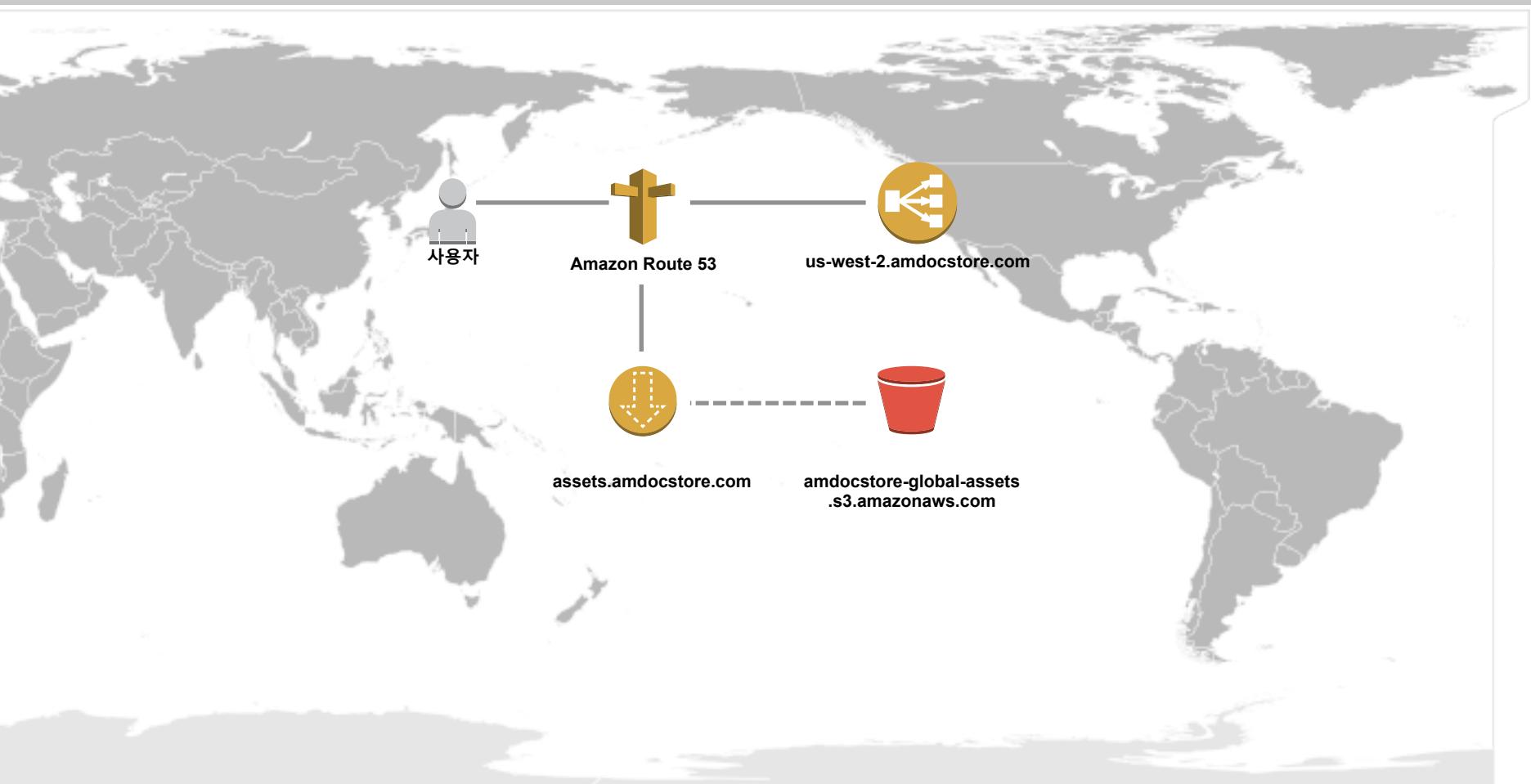
CloudFront는 S3 내 정적 웹 자산을 엣지 로케이션의 글로벌 네트워크를 이용해 짧은 지연 시간 내에 최종 사용자에게 전송합니다.

DocStore



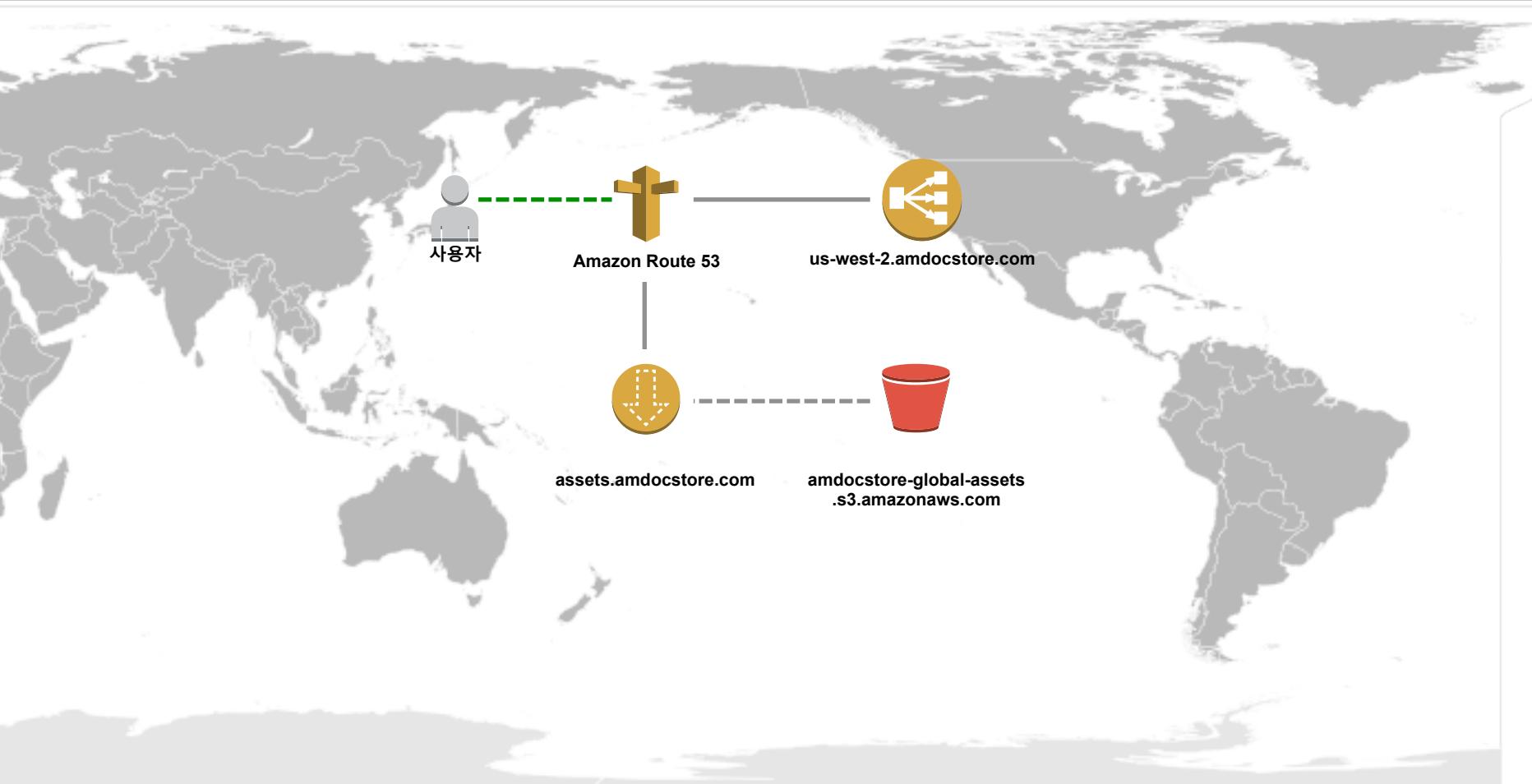
일본 도쿄의 사용자가 어떻게 [http://
assets.amdocstore.com/css/style.css](http://assets.amdocstore.com/css/style.css)에 액세스하는지 살
펴보겠습니다.

DocStore



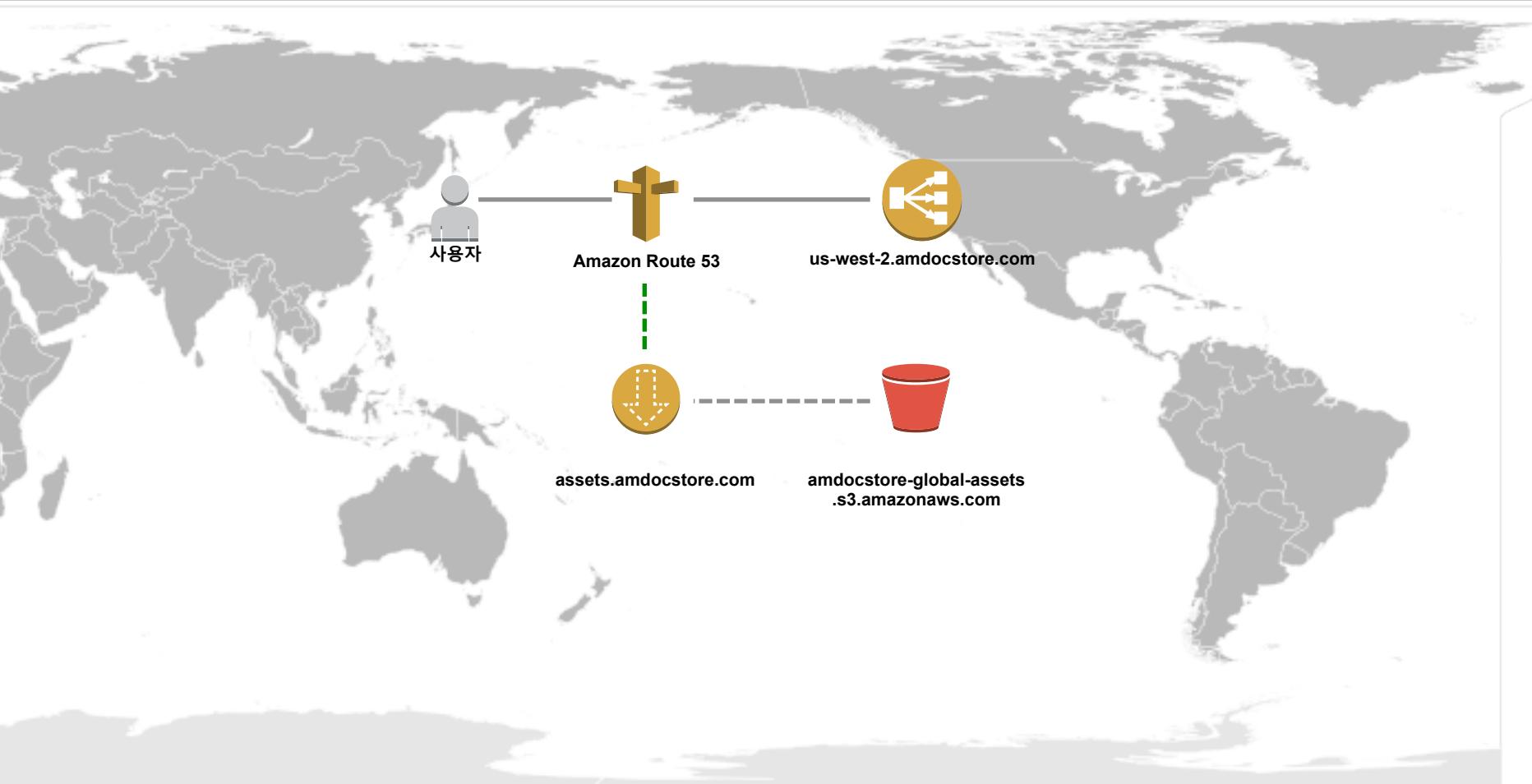
assets.amdocstore.com에 대한 모든 사용자 요청은 자동으로
가장 가까운 엣지 로케이션으로 라우팅됩니다.

DocStore



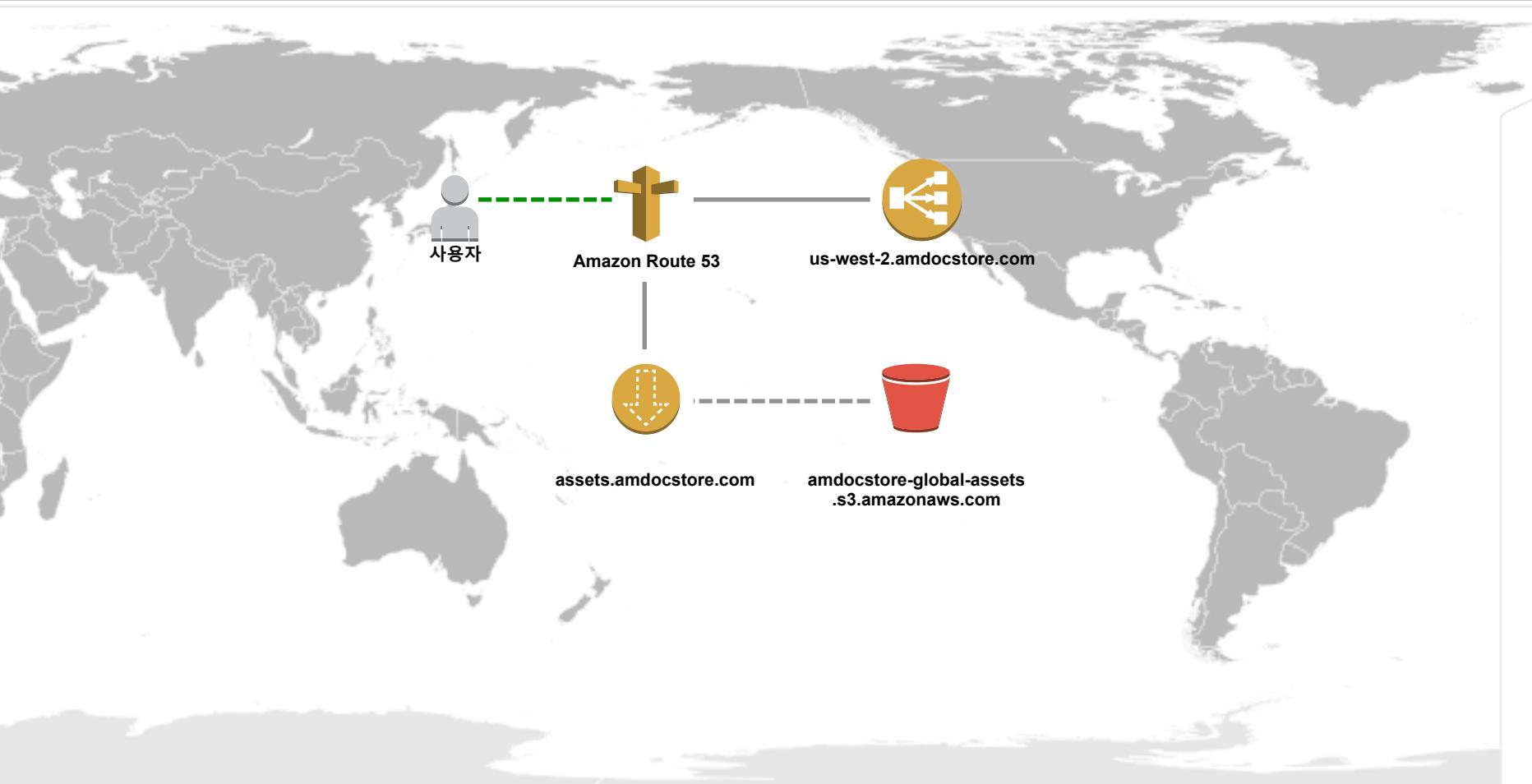
assets.amdocstore.com에 대한 모든 사용자 요청은 자동으로
가장 가까운 엣지 로케이션으로 라우팅됩니다.

DocStore



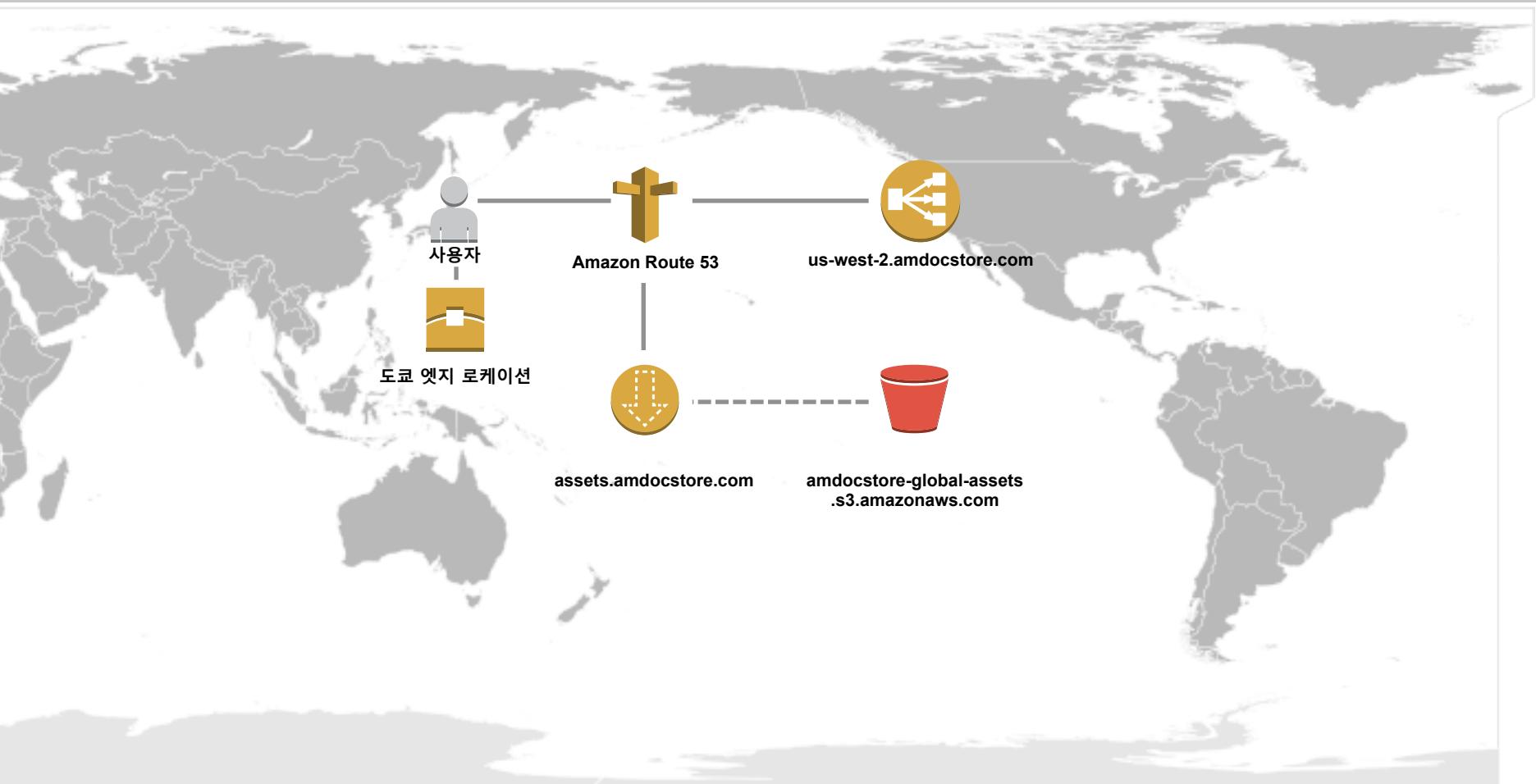
assets.amdocstore.com에 대한 모든 사용자 요청은 자동으로
가장 가까운 엣지 로케이션으로 라우팅됩니다.

DocStore



assets.amdocstore.com에 대한 모든 사용자 요청은 자동으로
가장 가까운 엣지 로케이션으로 라우팅됩니다.

DocStore



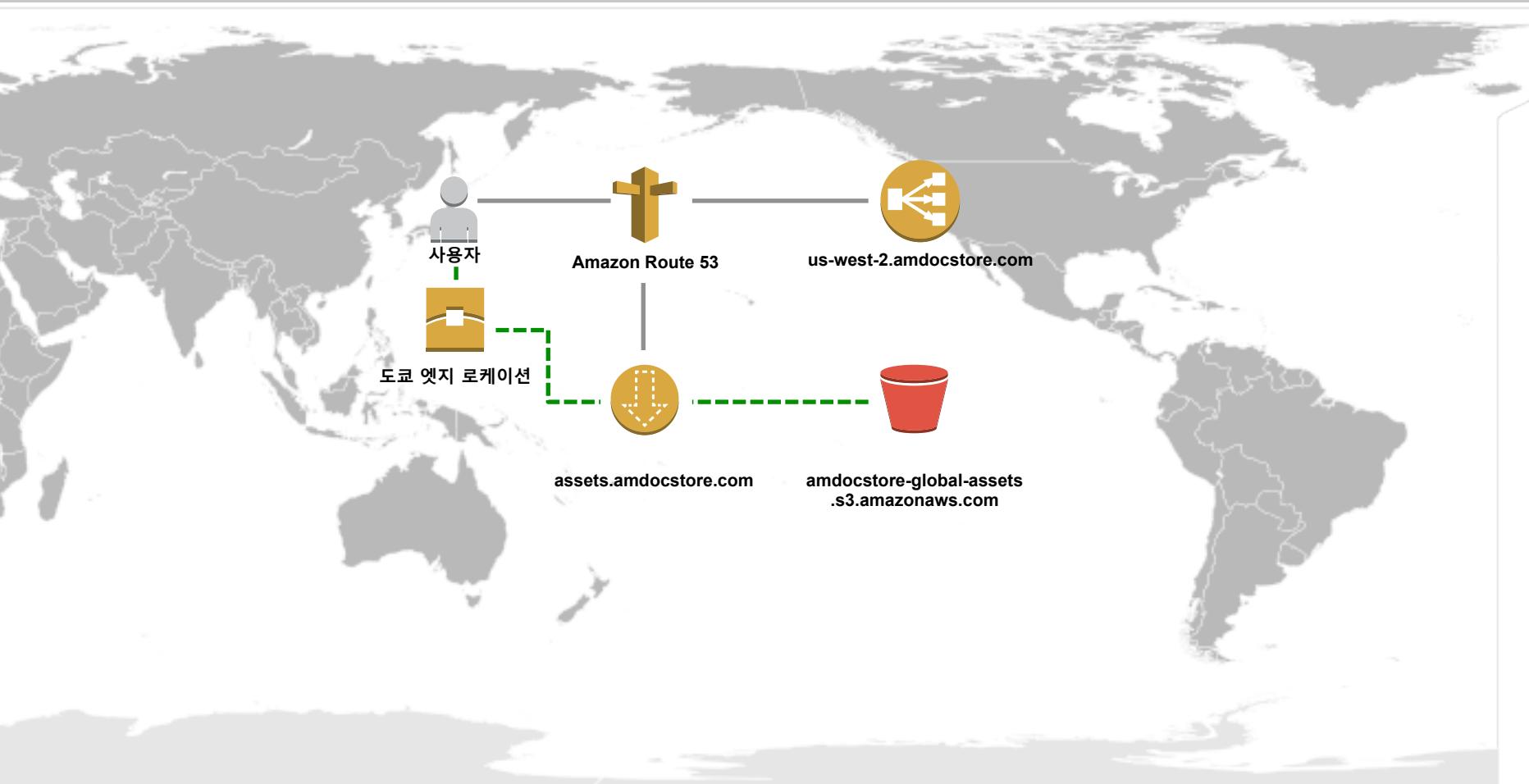
요청된 파일이 캐시에 있는 경우에는 일본 도쿄 엣지로부터
제공됩니다.

DocStore



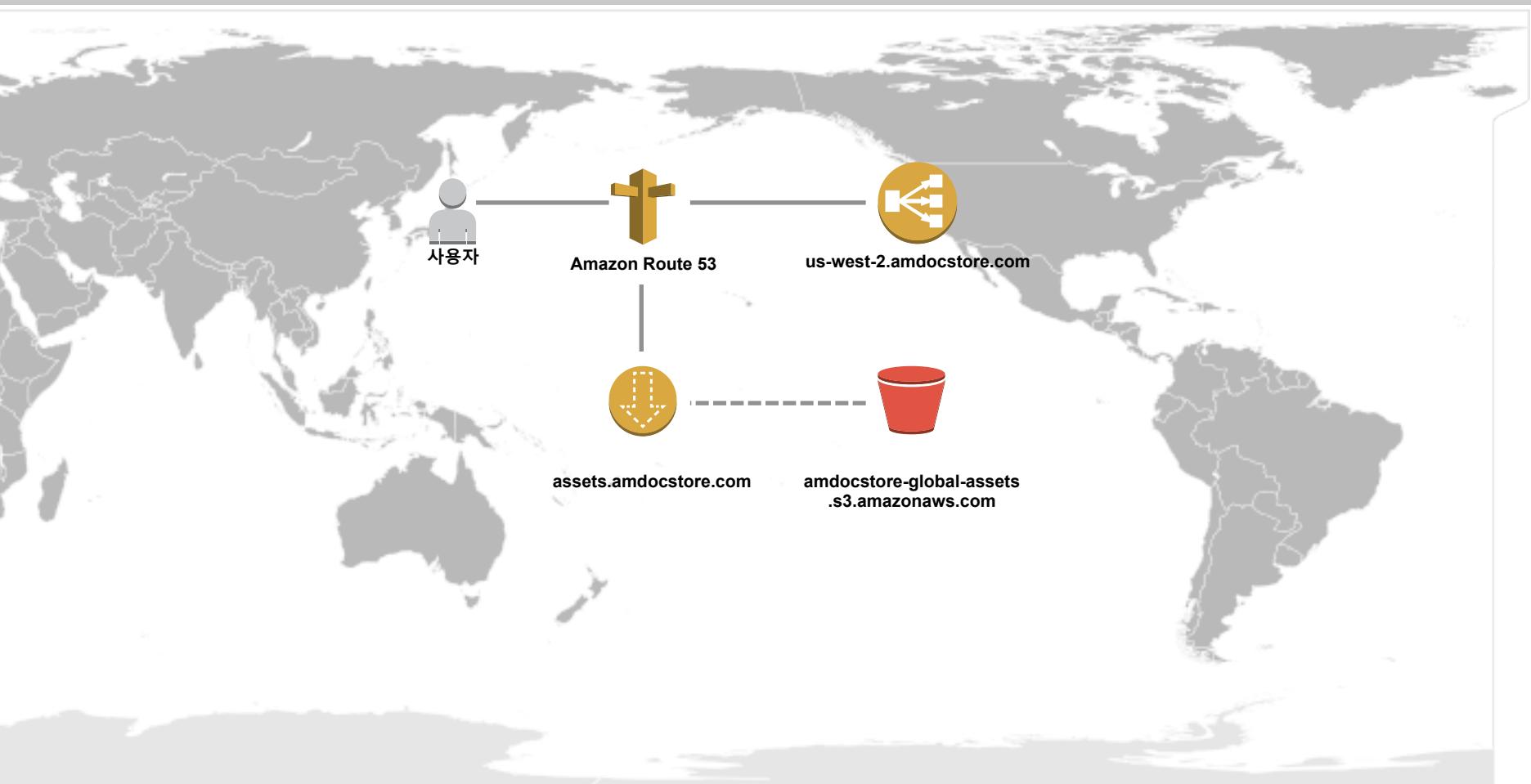
요청된 파일이 캐시에 있는 경우에는 일본 도쿄 엣지로부터 제공됩니다. 아니면 오리진으로부터 다운로드되며 추후 요청을 위해 엣지에 캐싱됩니다.

DocStore



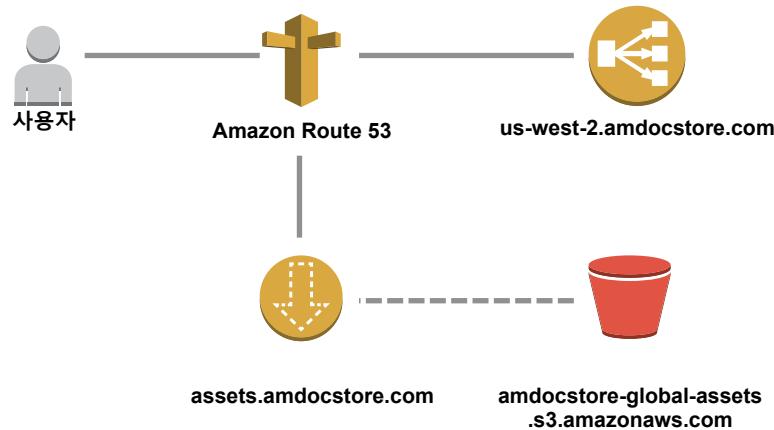
us-west-2 내에서의 애플리케이션 배치로 돌아가겠습니다.

DocStore



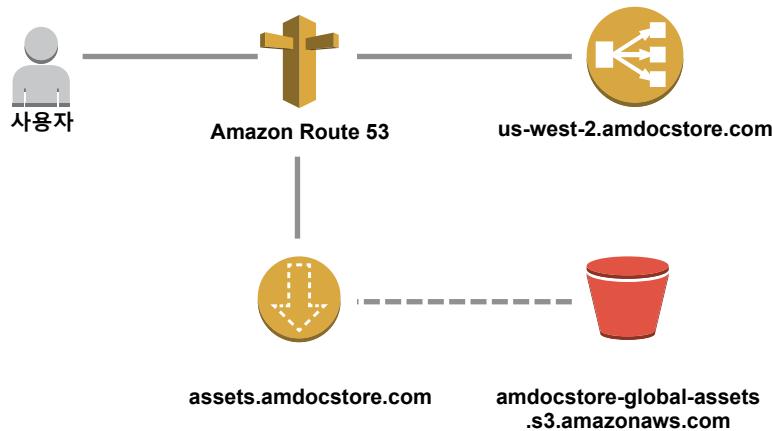
us-west-2 내에서의 애플리케이션 배치로 돌아가겠습니다.

DocStore



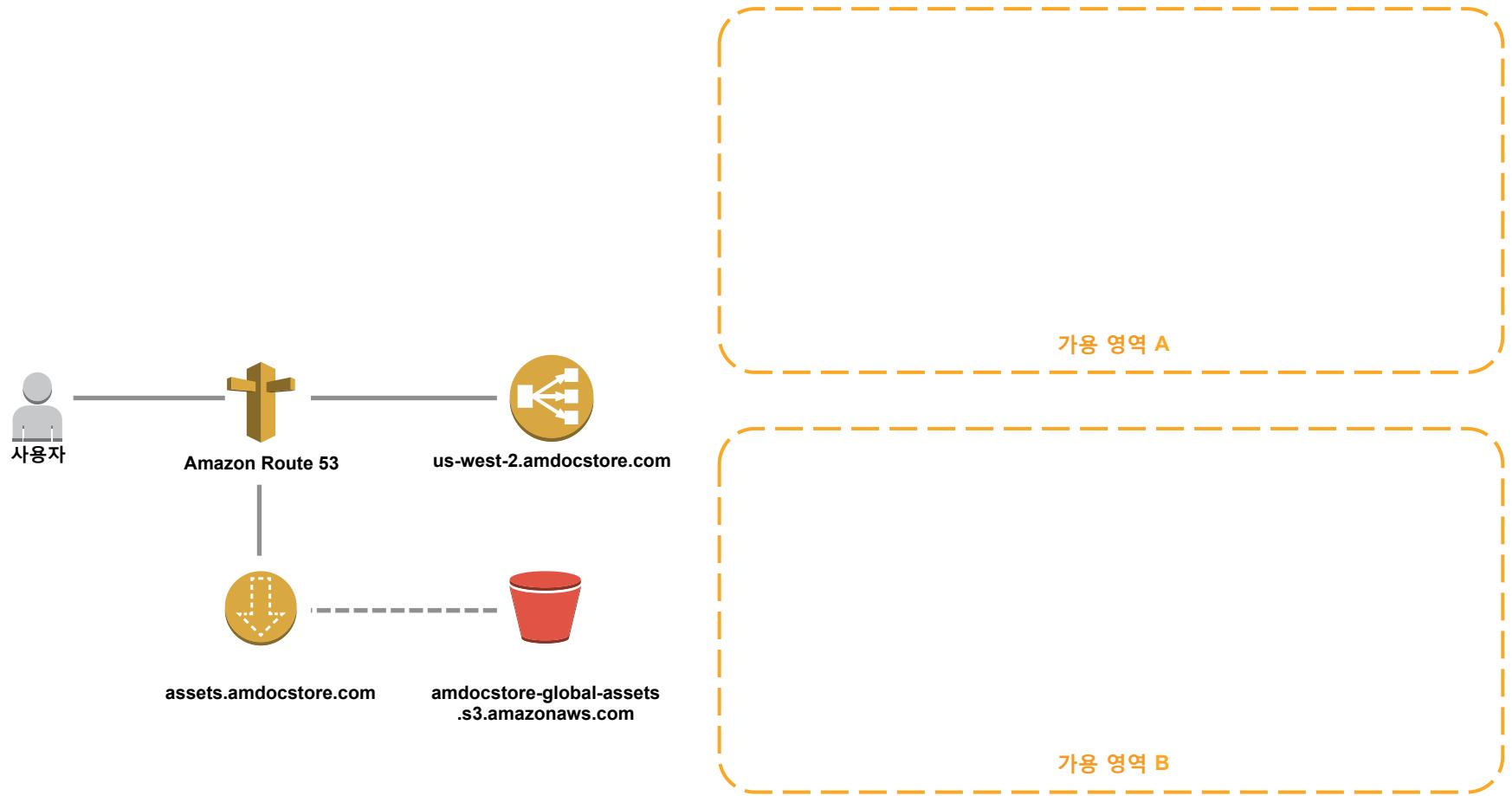
us-west-2 내에서의 애플리케이션 배치로 돌아가겠습니다.
그리고 DocStore의 브라우저를 위한 구성 요소를 집중적으
로 살펴보겠습니다.

DocStore



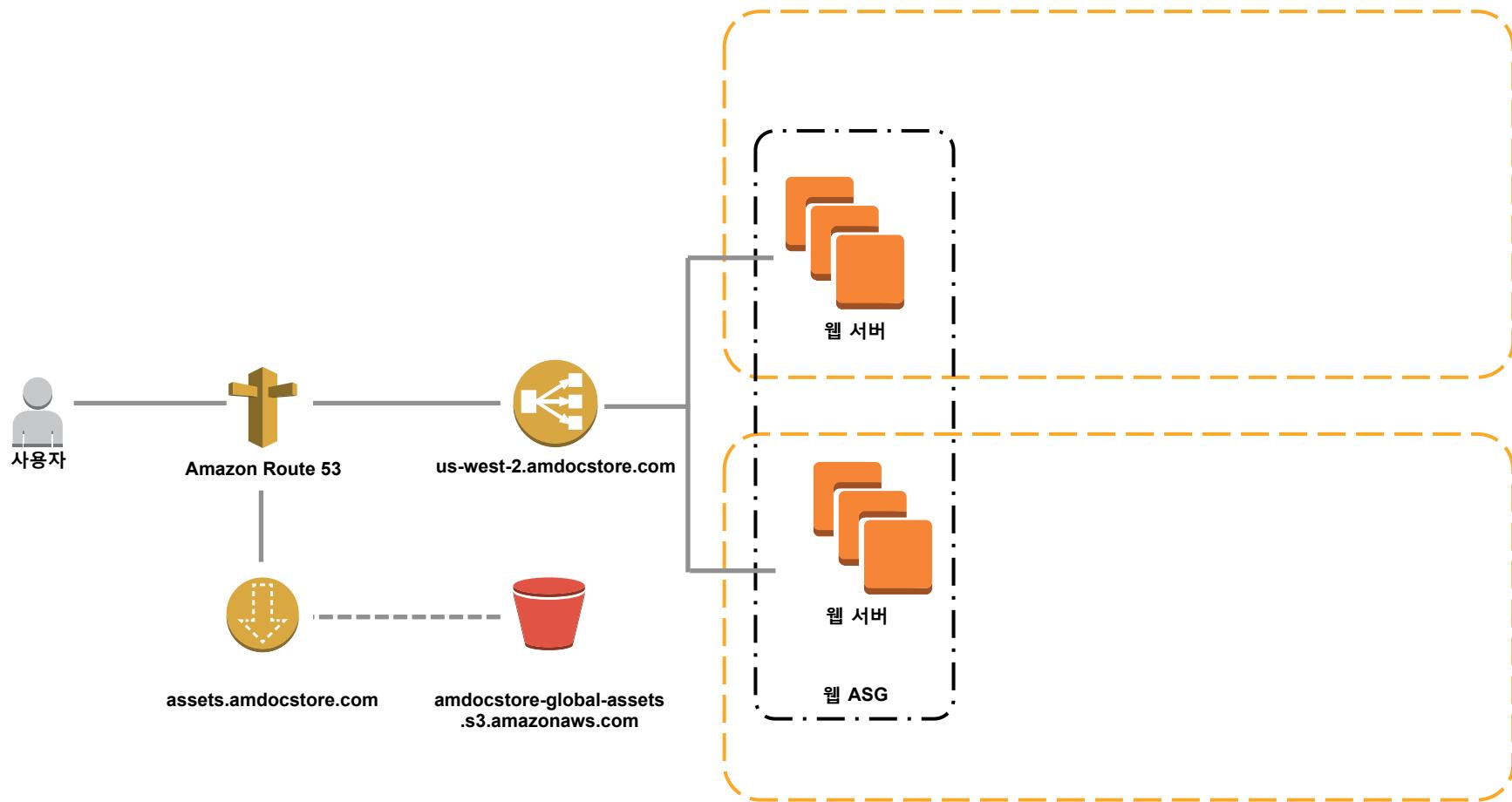
웹 서버는 Auto Scaling을 이용해 복수의 가용 영역에 배치됩니다.

DocStore



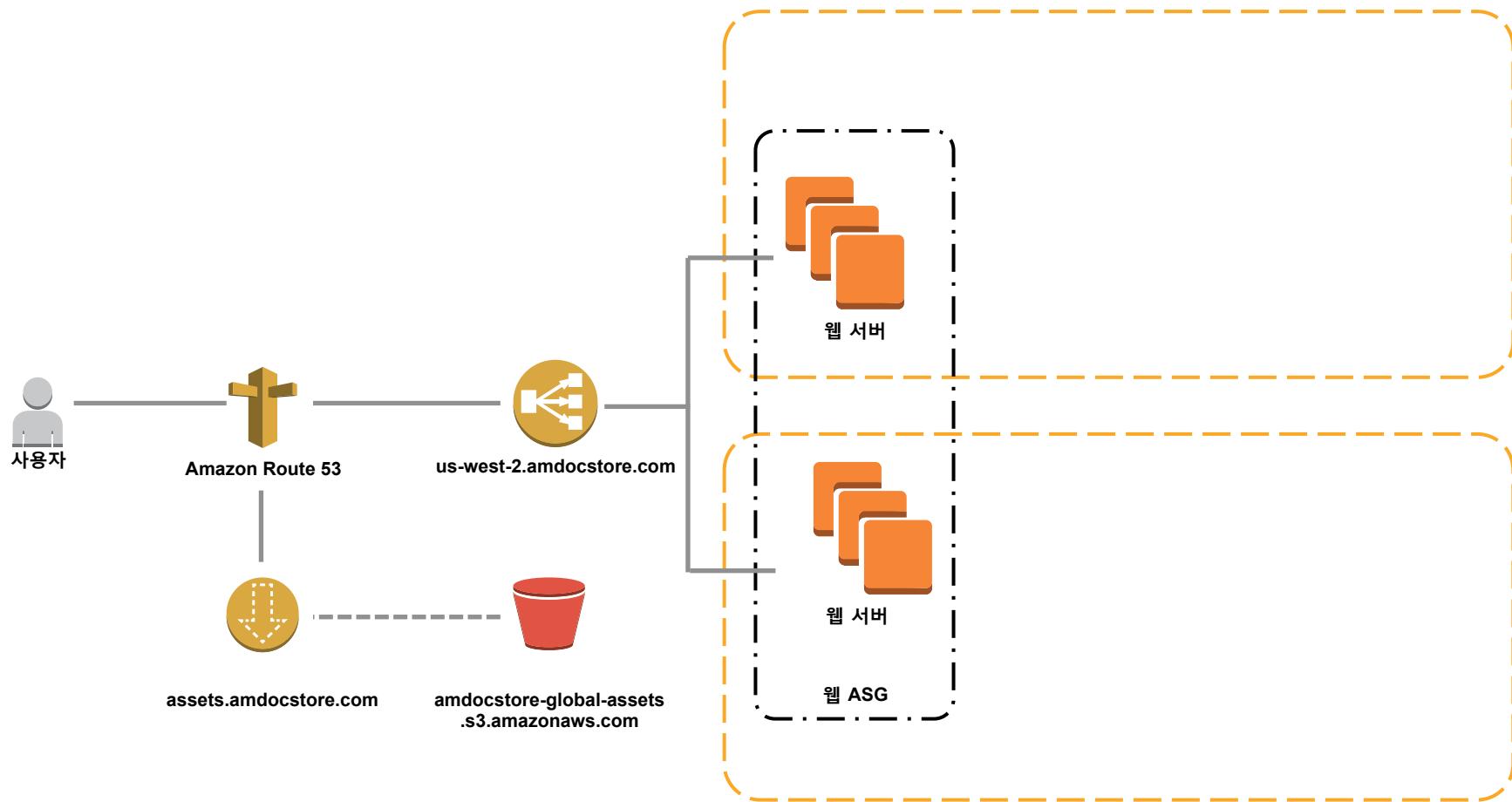
웹 서버는 Auto Scaling을 이용해 복수의 가용 영역에 배치됩니다.

DocStore



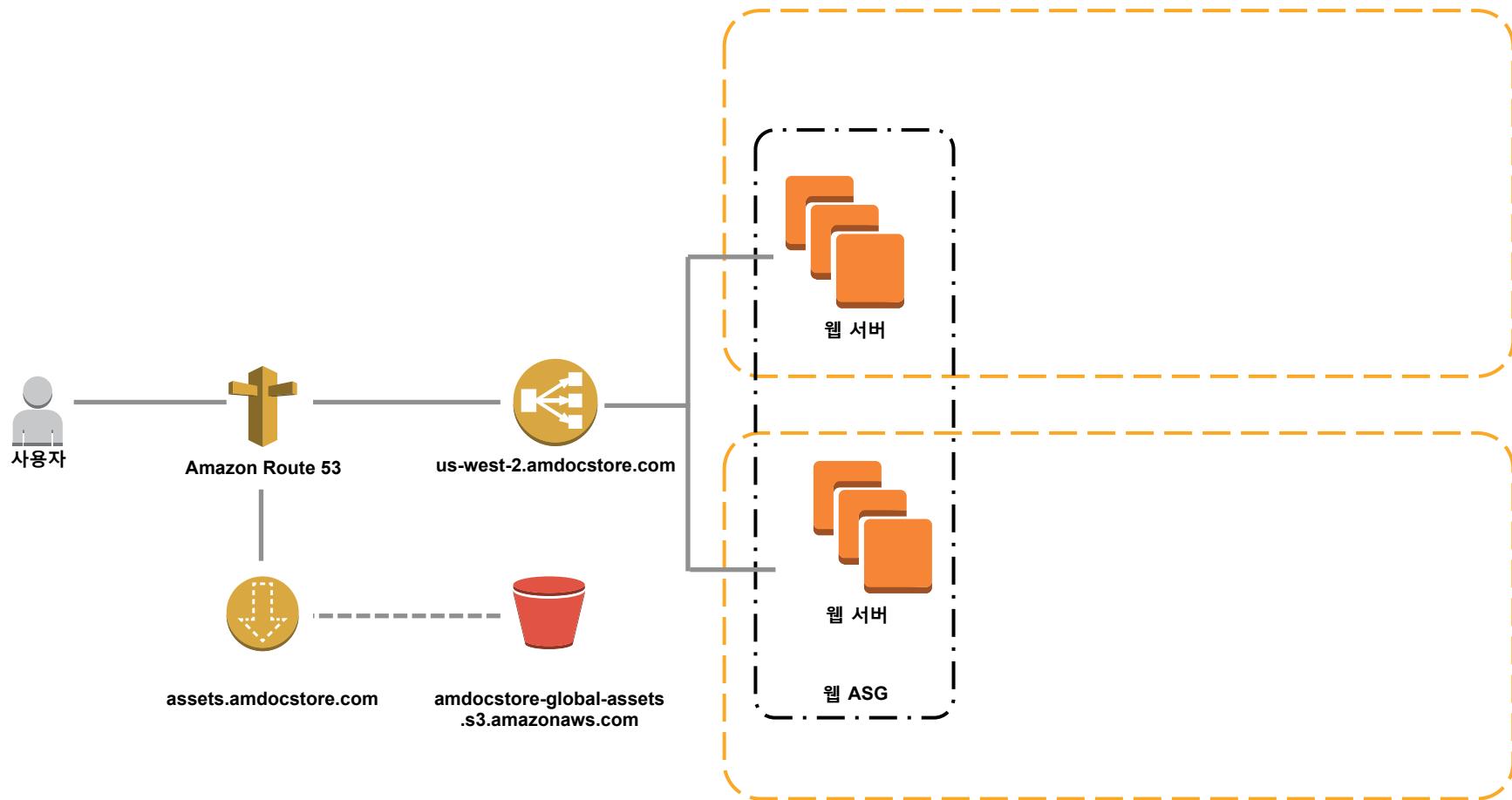
웹 서버는 사용자가 웹 브라우저를 통해 액세스하는 DocStore 인터페이스를 호스팅합니다.

DocStore



웹 서버는 사용자 인터페이스/뷰를 렌더링하기 위해 루비/레일, PHP, 파이썬 또는 다른 도구를 구동할 것입니다.

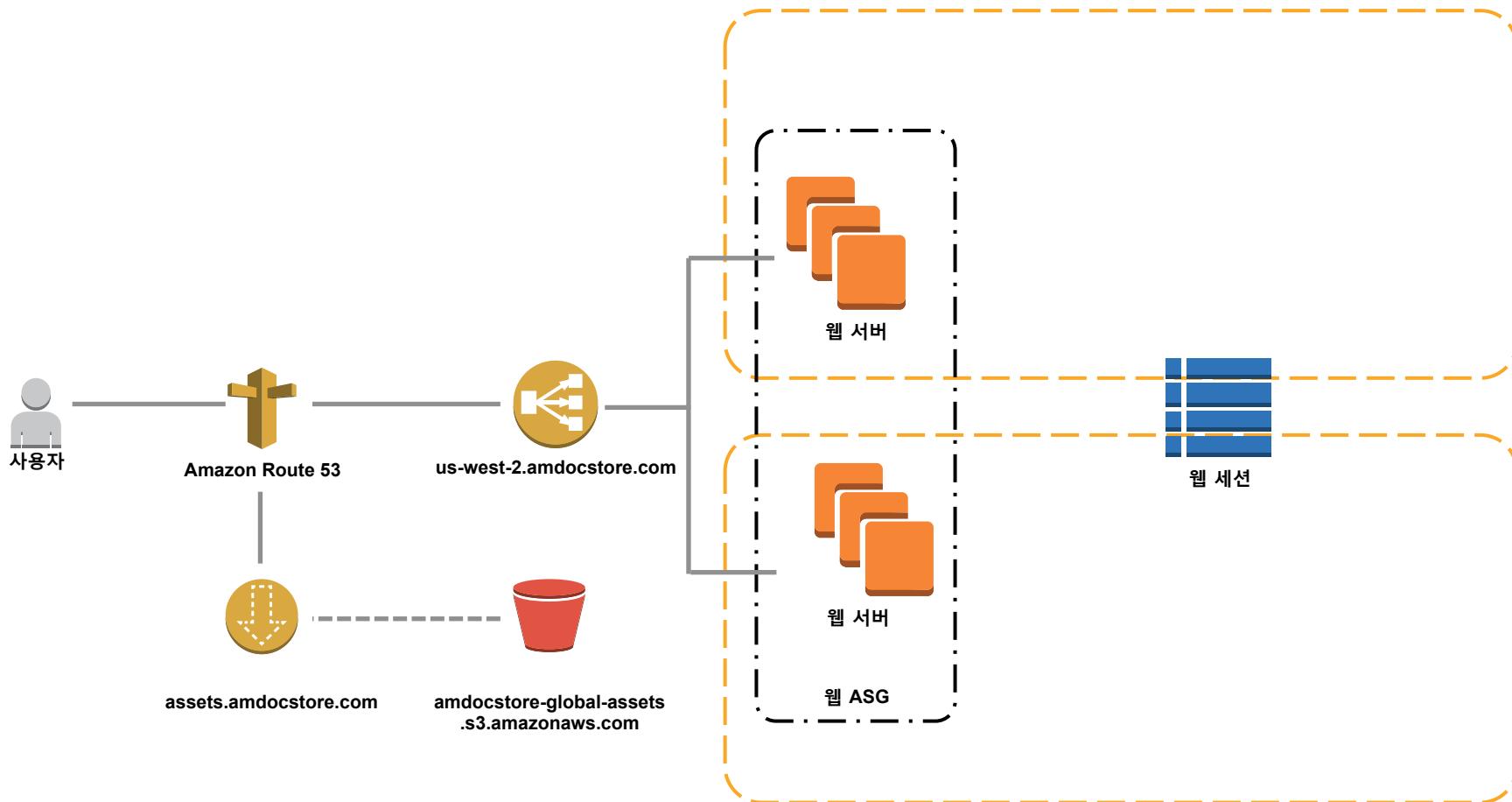
DocStore



세션 상태는 웹 계층 외부의 **DynamoDB** 테이블
내에 저장됩니다. DynamoDB는 지역 범위
서비스이므로 자동으로 여러 AZ에 배치됩니다.



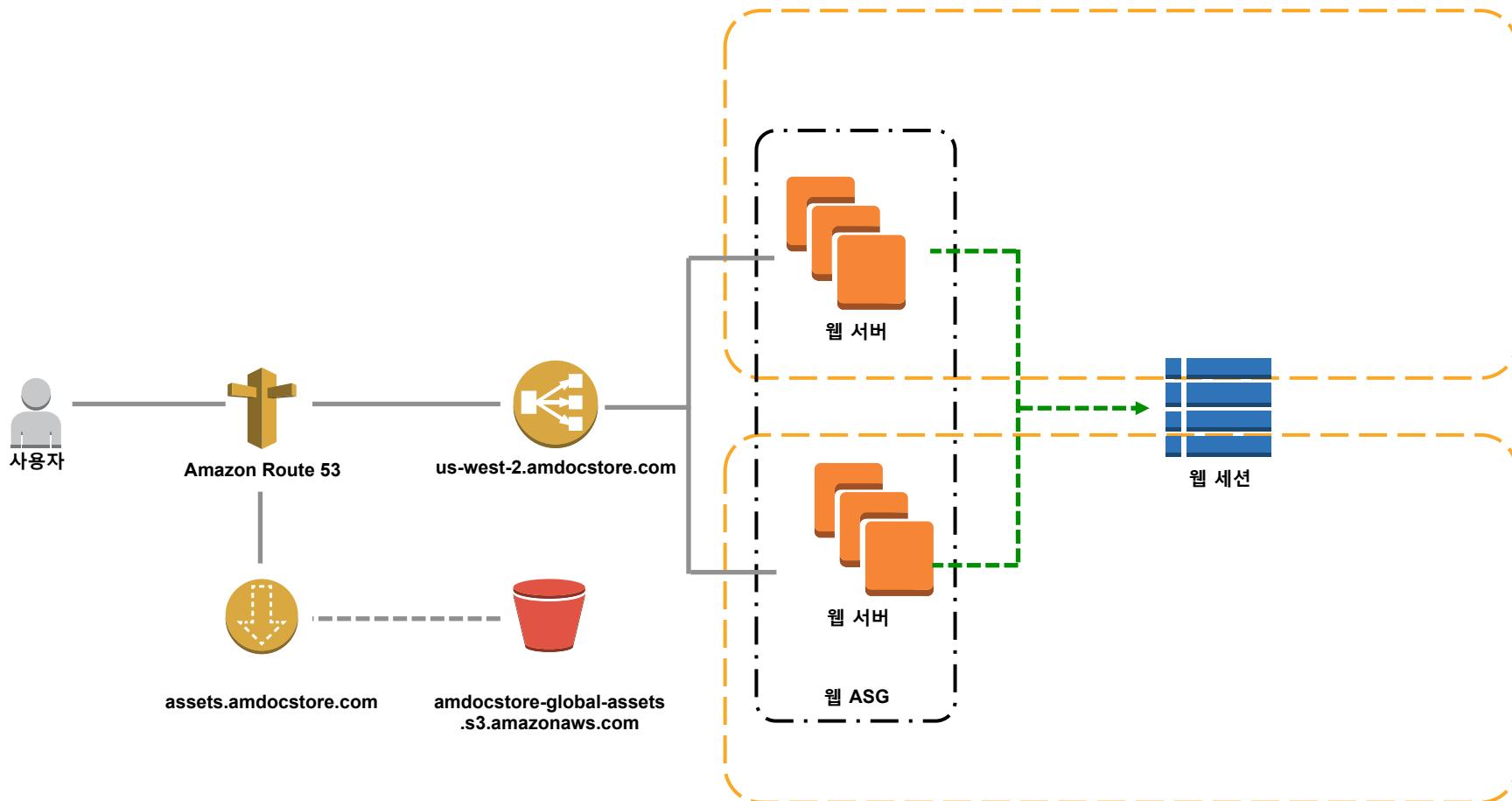
DocStore



세션 상태는 웹 계층 외부의 **DynamoDB** 테이블
내에 저장됩니다. DynamoDB는 지역 범위
서비스이므로 자동으로 여러 AZ에 배치됩니다.

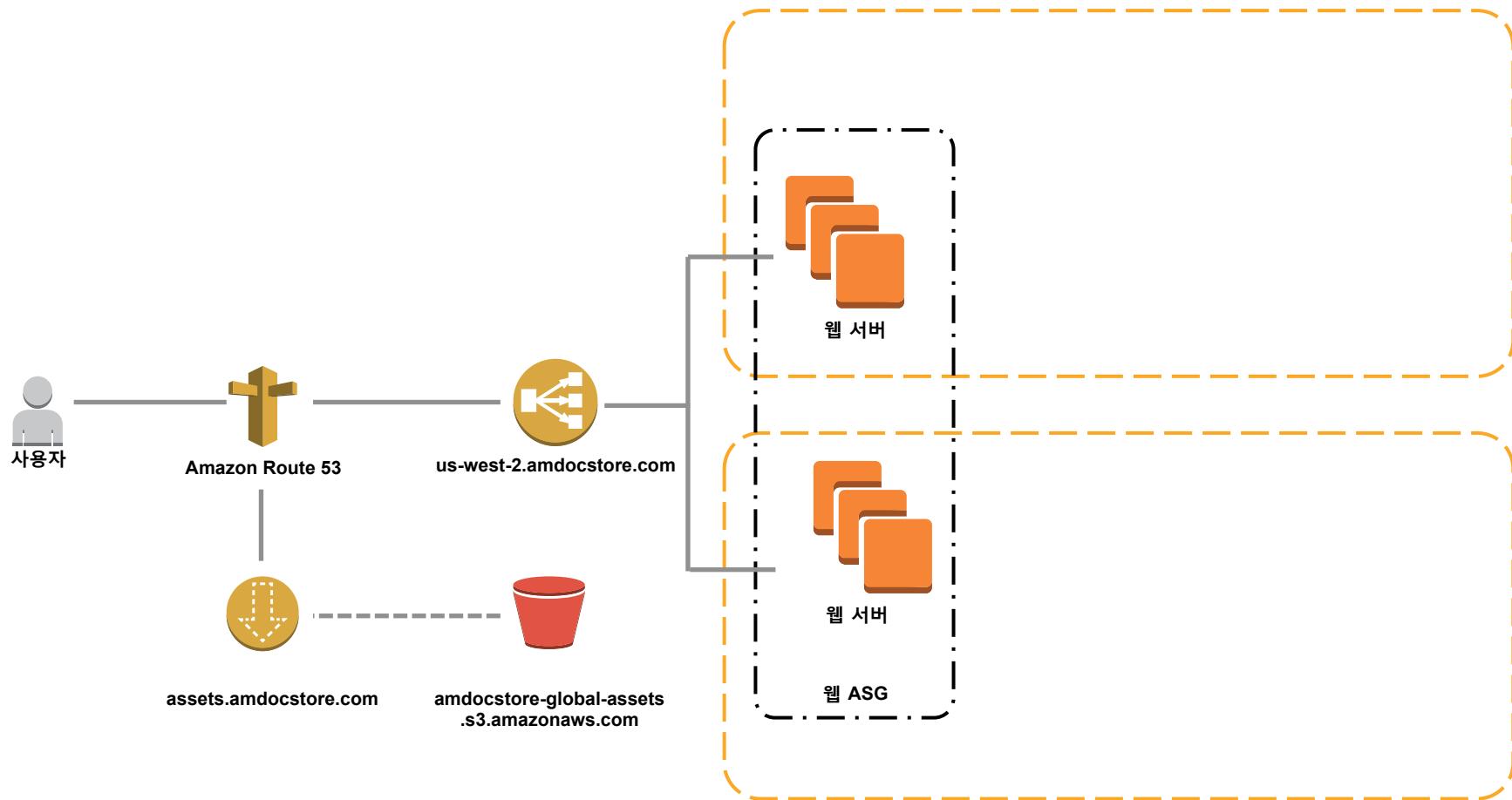


DocStore



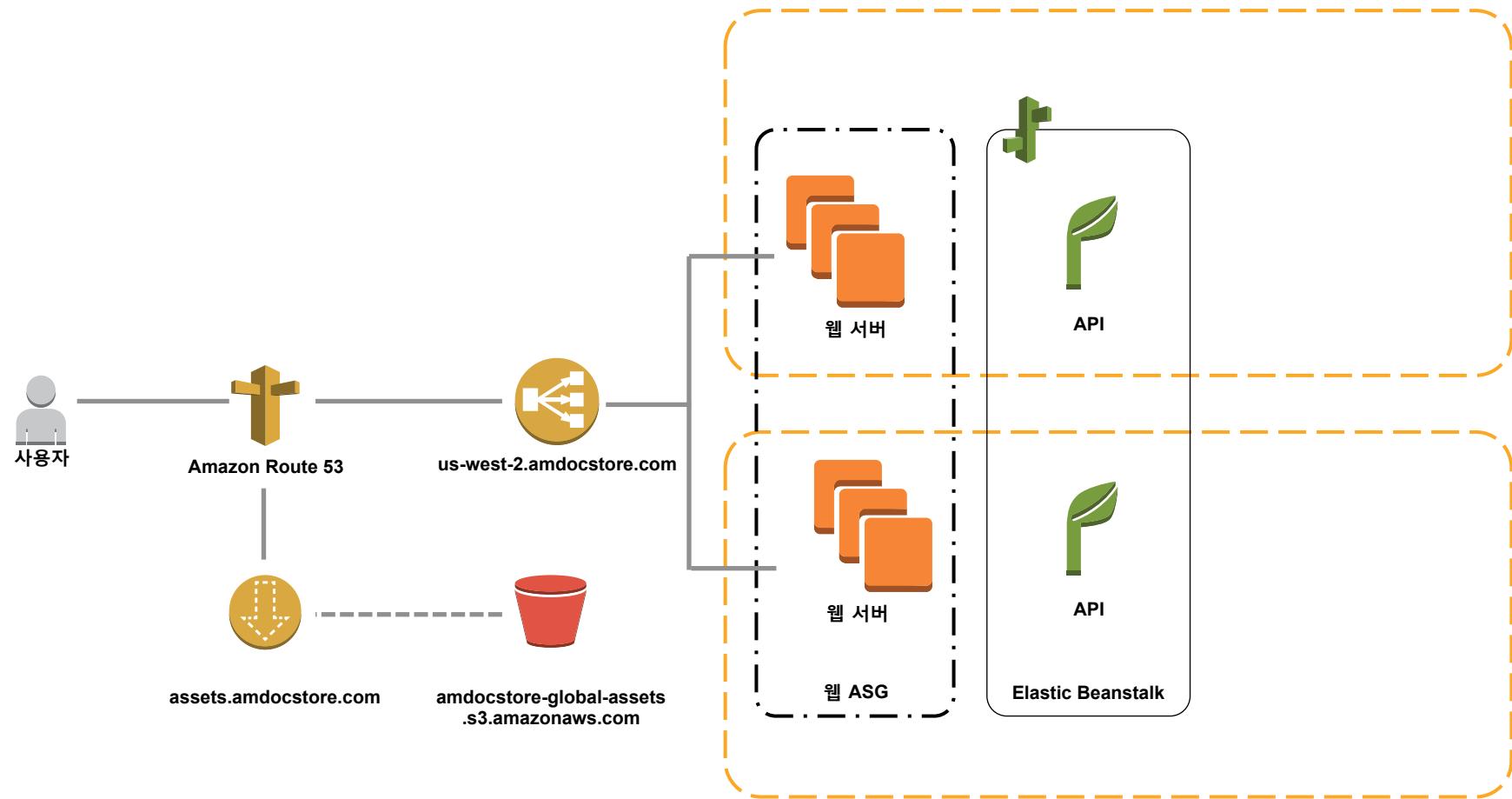
핵심 기능은 **DocStore API**에 의해 제공되고, 웹 서버는 이 API와 연동합니다.

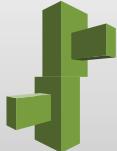
DocStore



DocStore API는 **Elastic Beanstalk** 컨테이너 애플리케이션으로 배치됩니다.

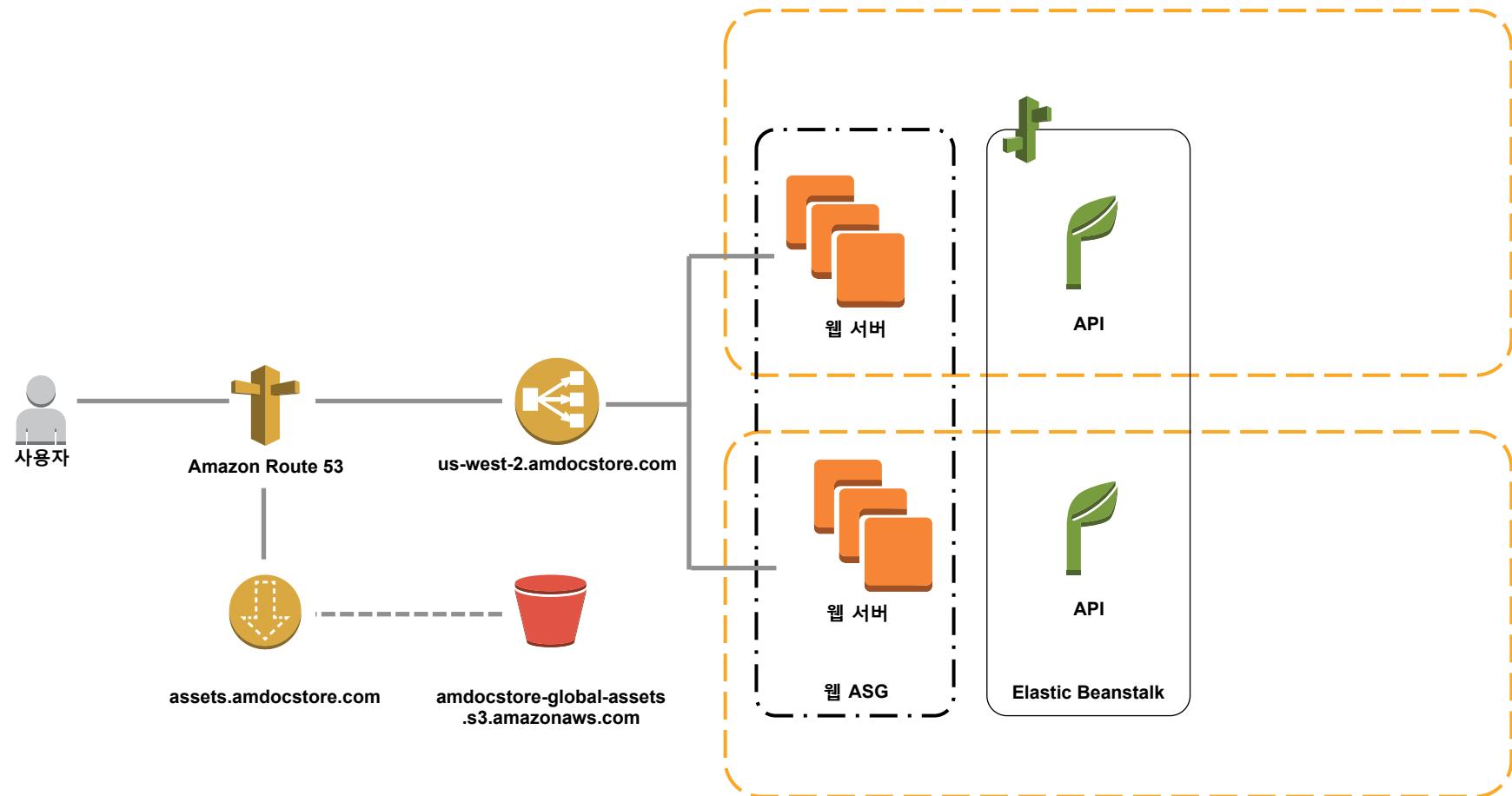
DocStore

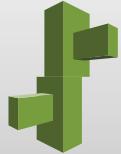




Elastic Beanstalk는 다중 AZ 배치, Auto Scaling 및
로드밸런싱을 제공합니다.

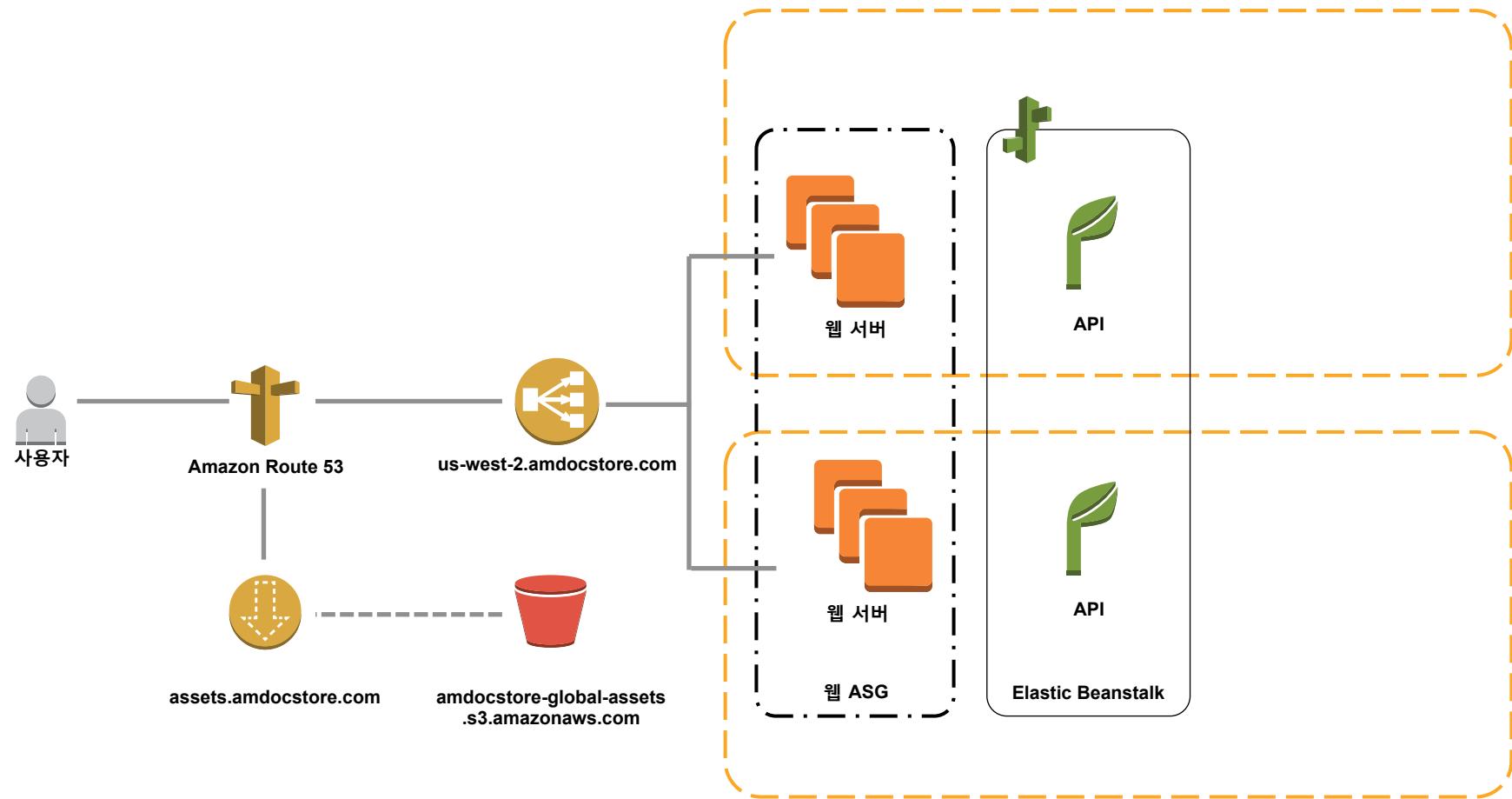
DocStore





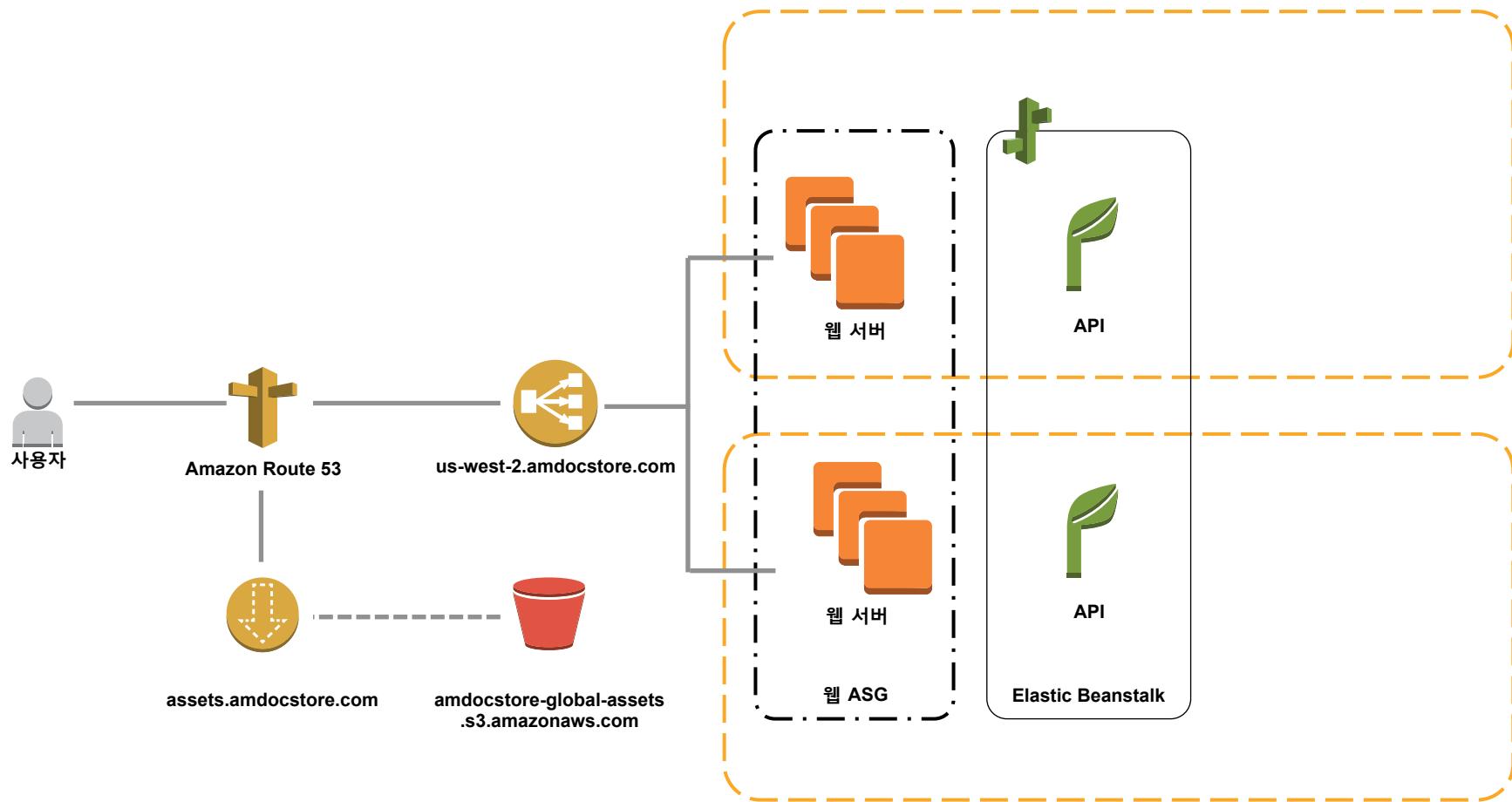
Elastic Beanstalk은 자바, 닷넷, 파이썬, PHP, 루비 및 node.js 애플리케이션을 지원합니다.

DocStore



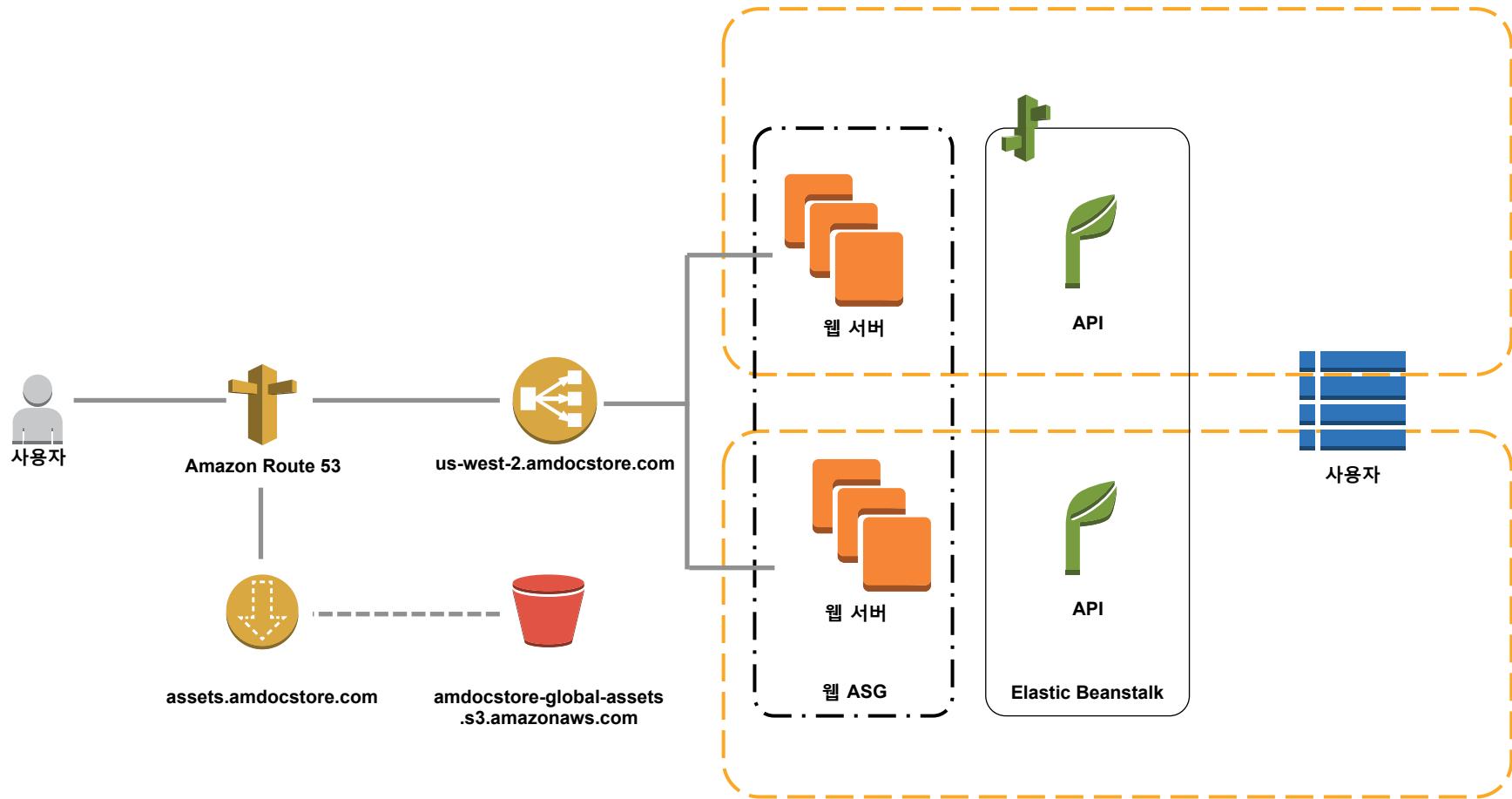
API는 데이터 유형에 따라 다양한 데이터 스토어를 사용합니다.

DocStore



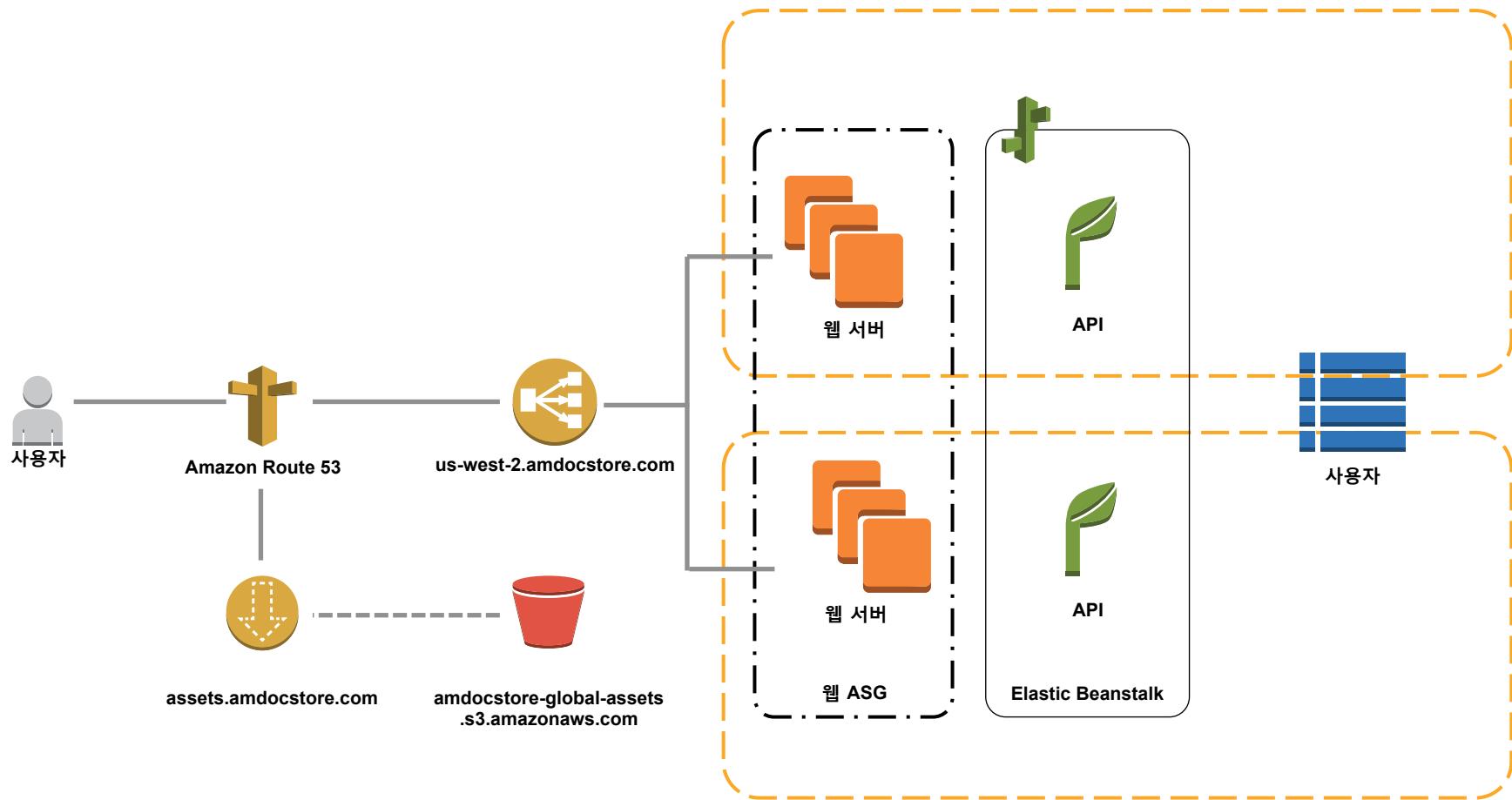
사용자 계정 정보는 DynamoDB 테이블에 저장됩니다.

DocStore



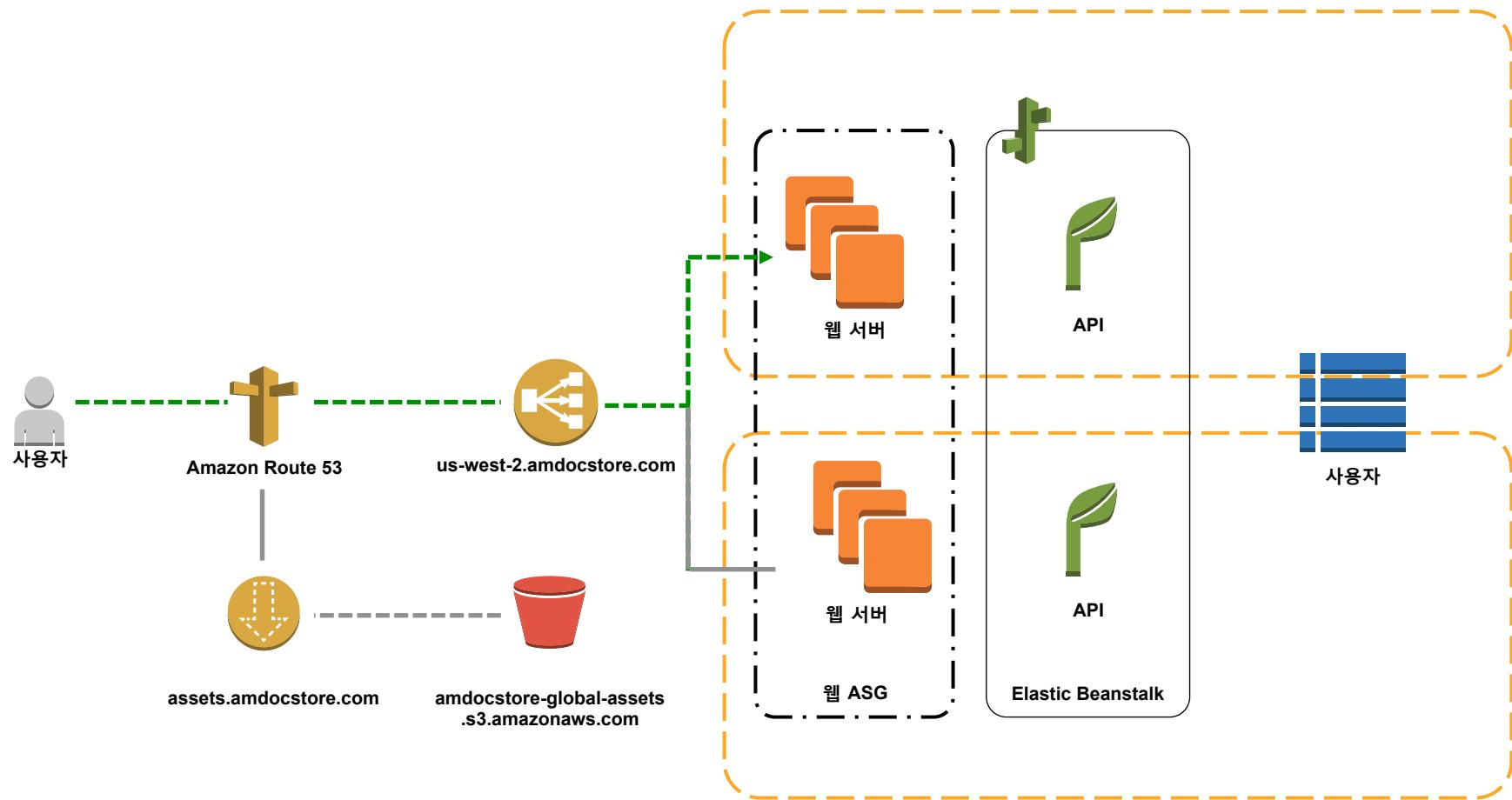
새로운 사용자가 DocStore에 가입하면 어떻게 처리되는지
살펴보겠습니다.

DocStore



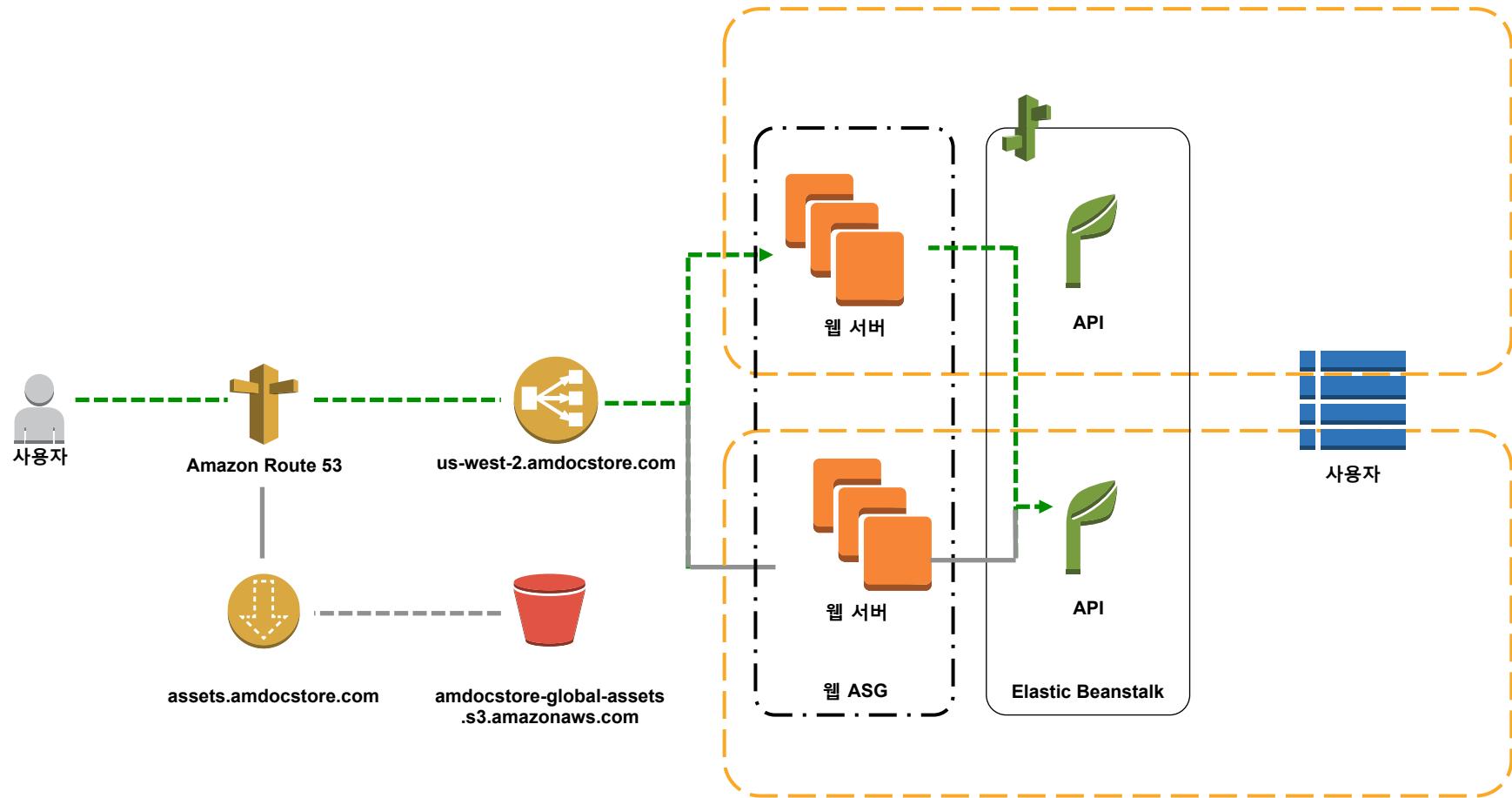
계정 생성: 사용자가 사용자 이름, 암호 등을 포함한 가입 양식을 웹 서버에 제출합니다.

DocStore



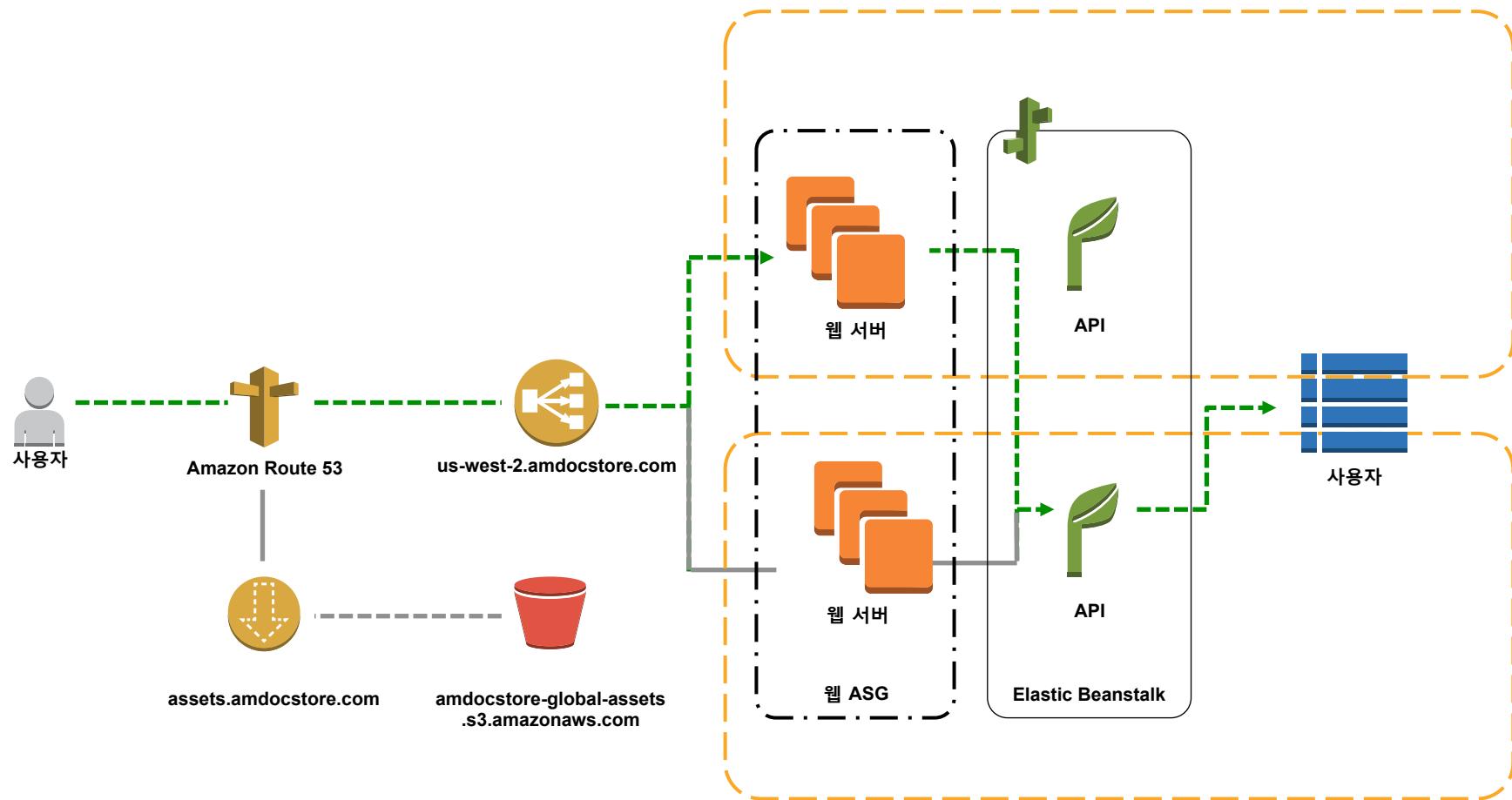
계정 생성: 웹 서버는 CreateAccount API를 호출합니다.

DocStore



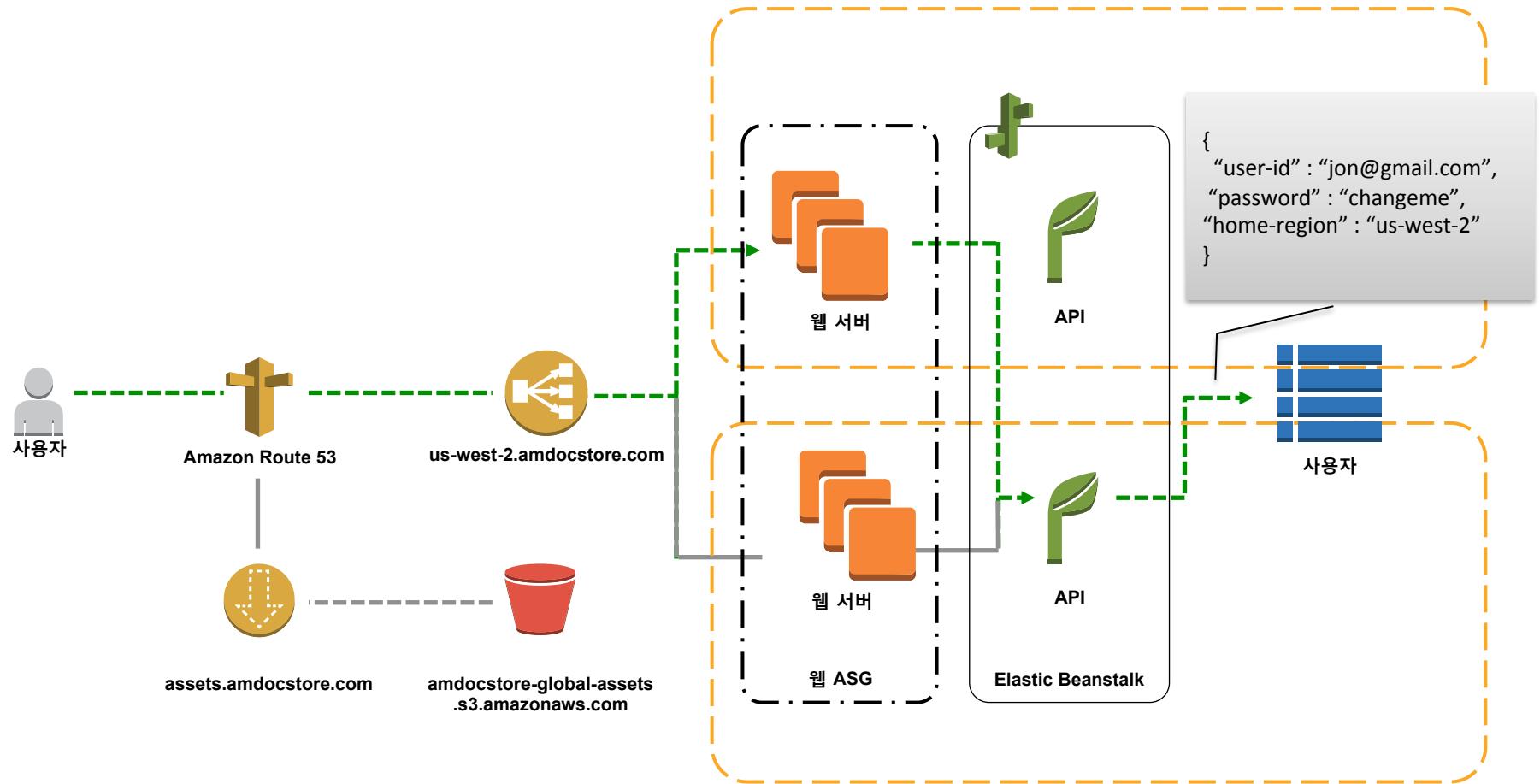
계정 생성: API는 DynamoDB 테이블에 새 항목을 생성합니다.

DocStore



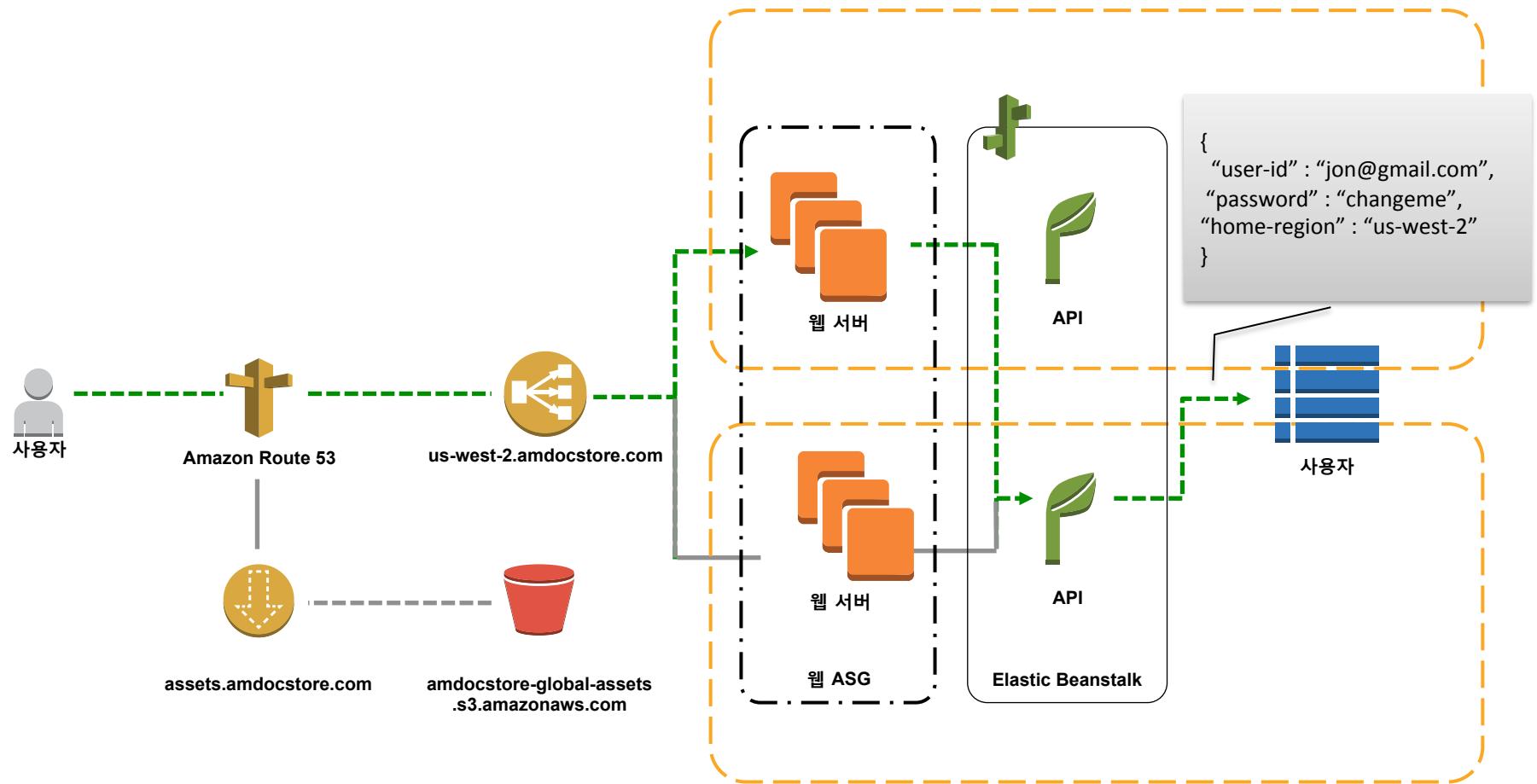
계정 생성: API는 데이터 스토어에 새 항목을 생성합니다.

DocStore



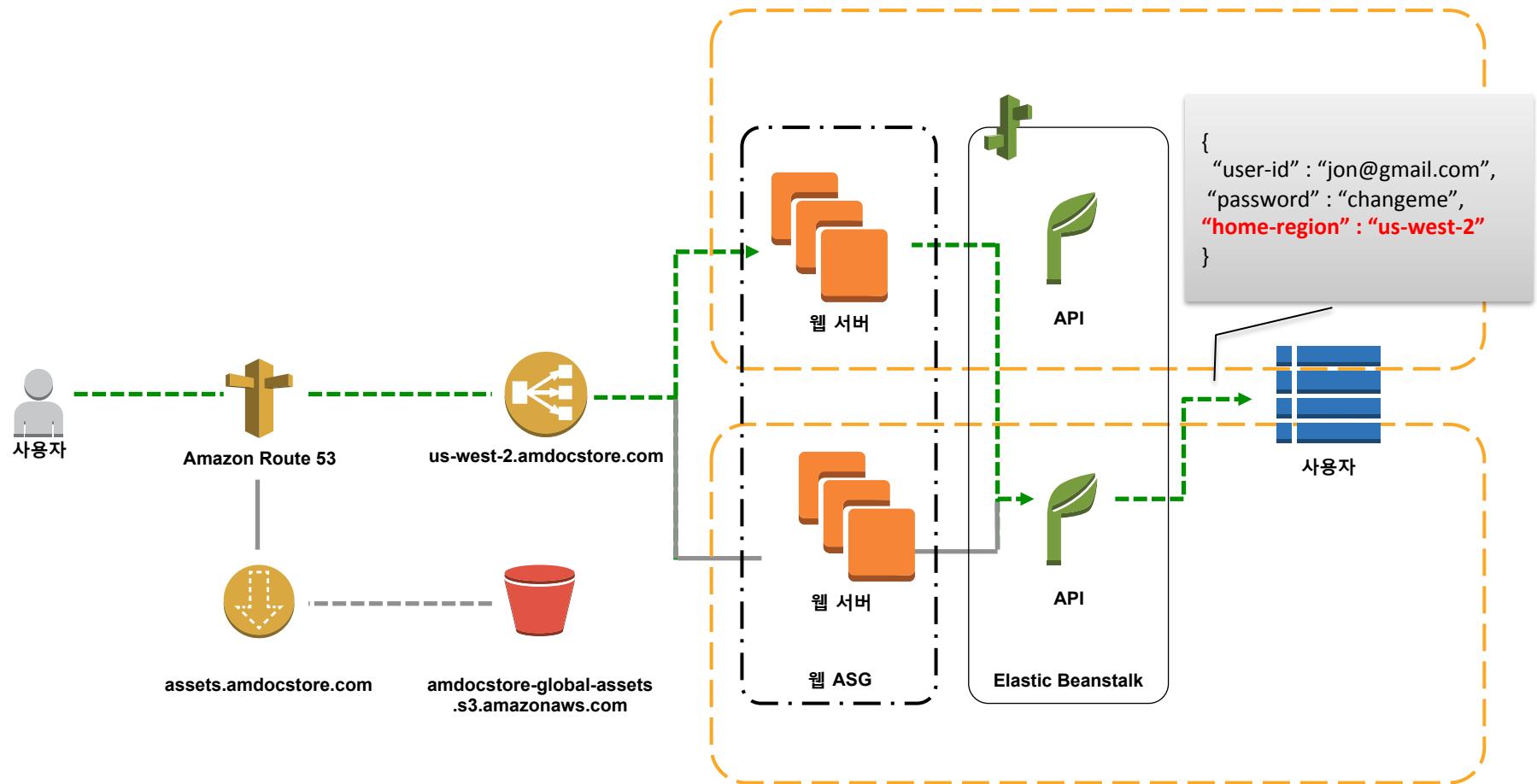
계정 생성: Elastic Beanstalk 내에 배치된 API 앱에는 현재 실행되는 지역을 인식할 수 있는 구성이 설정되어 있습니다.

DocStore



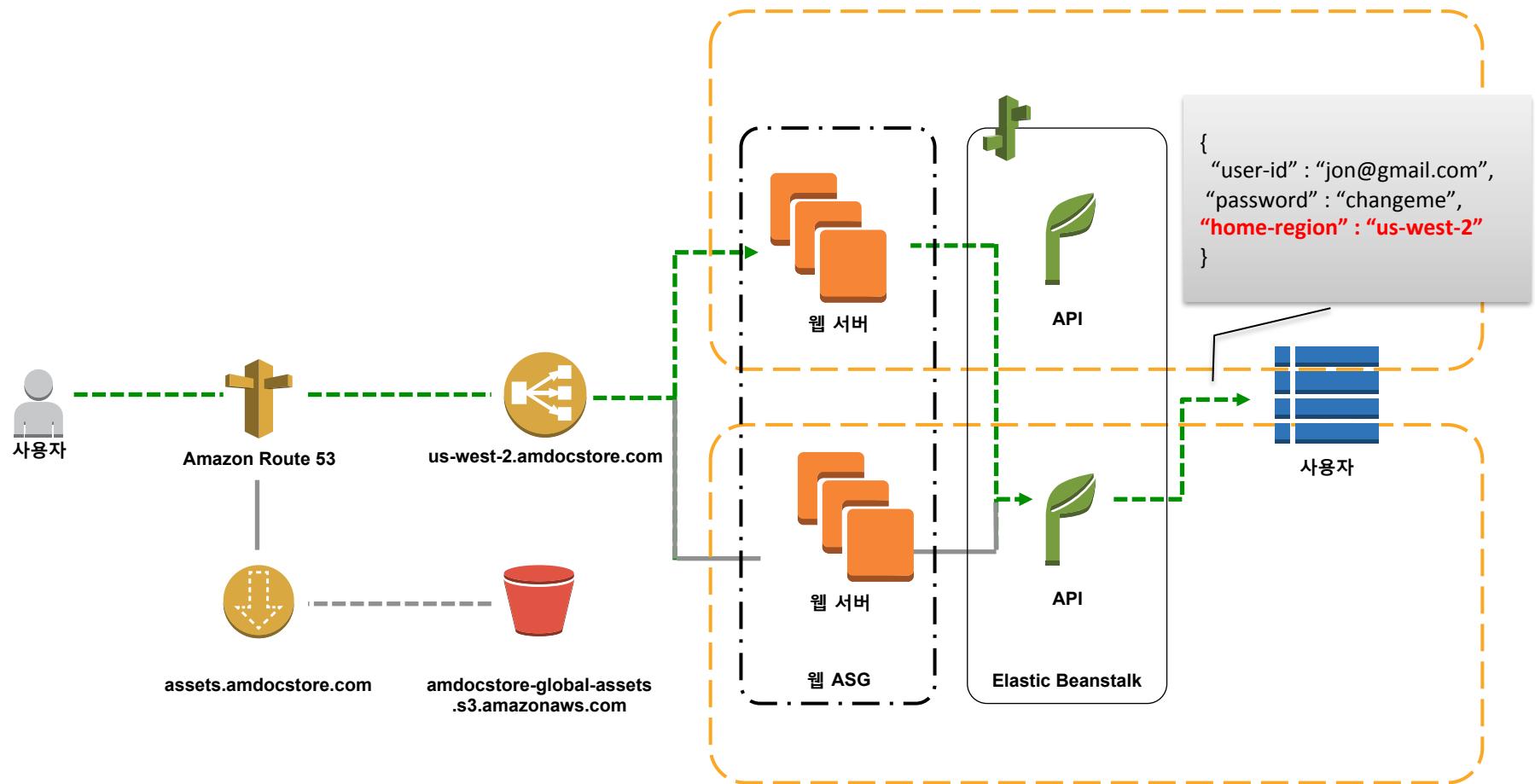
계정 생성: API는 표준 정보 외에도 사용자의 '홈 지역', 즉 사용자가 가입한 지역 정보를 저장합니다.

DocStore



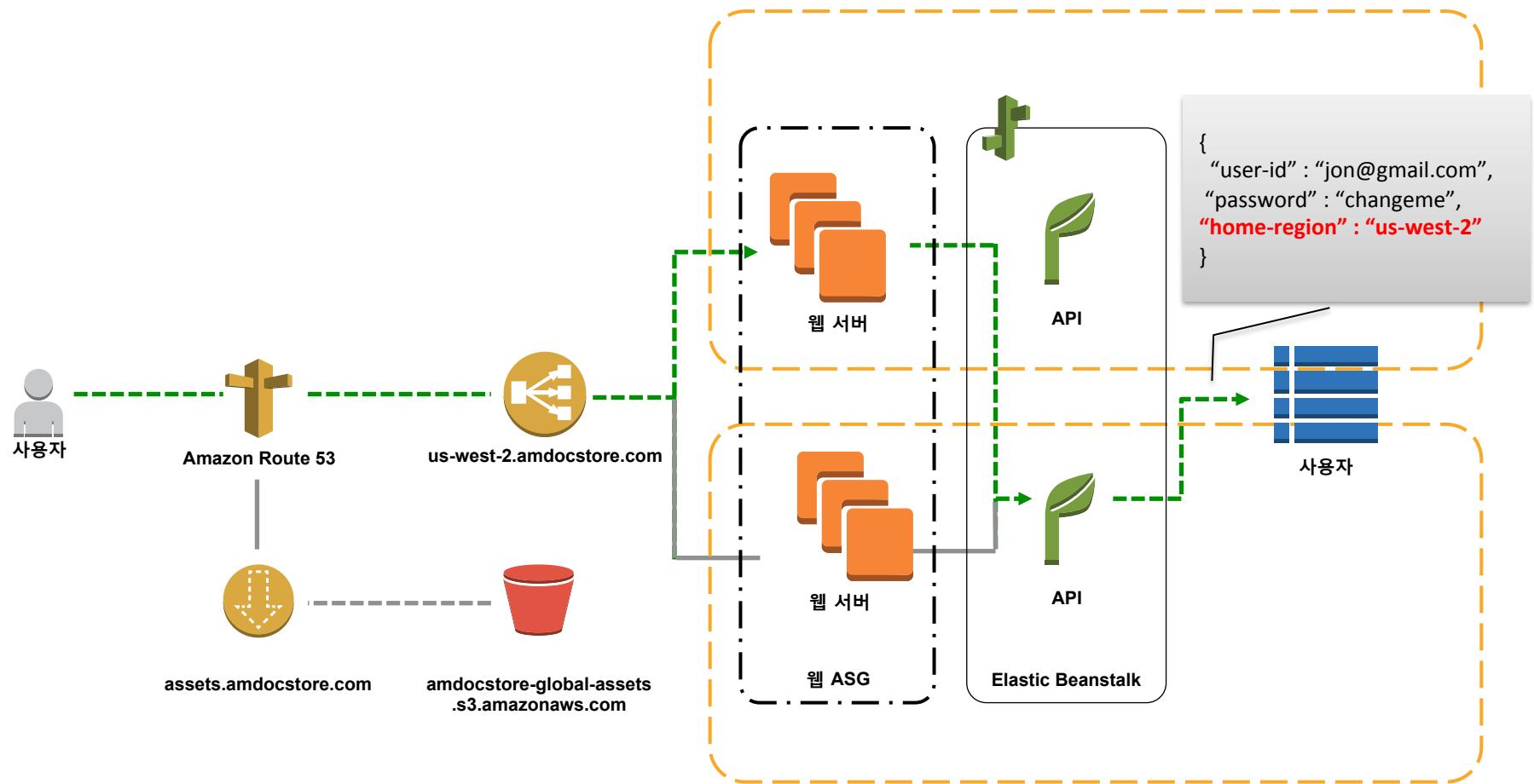
계정 생성: API는 시드니로 계정 데이터를 복제하여 다른 지역에서 액세스하는 사용자를 홈 지역으로 리디렉션합니다.

DocStore



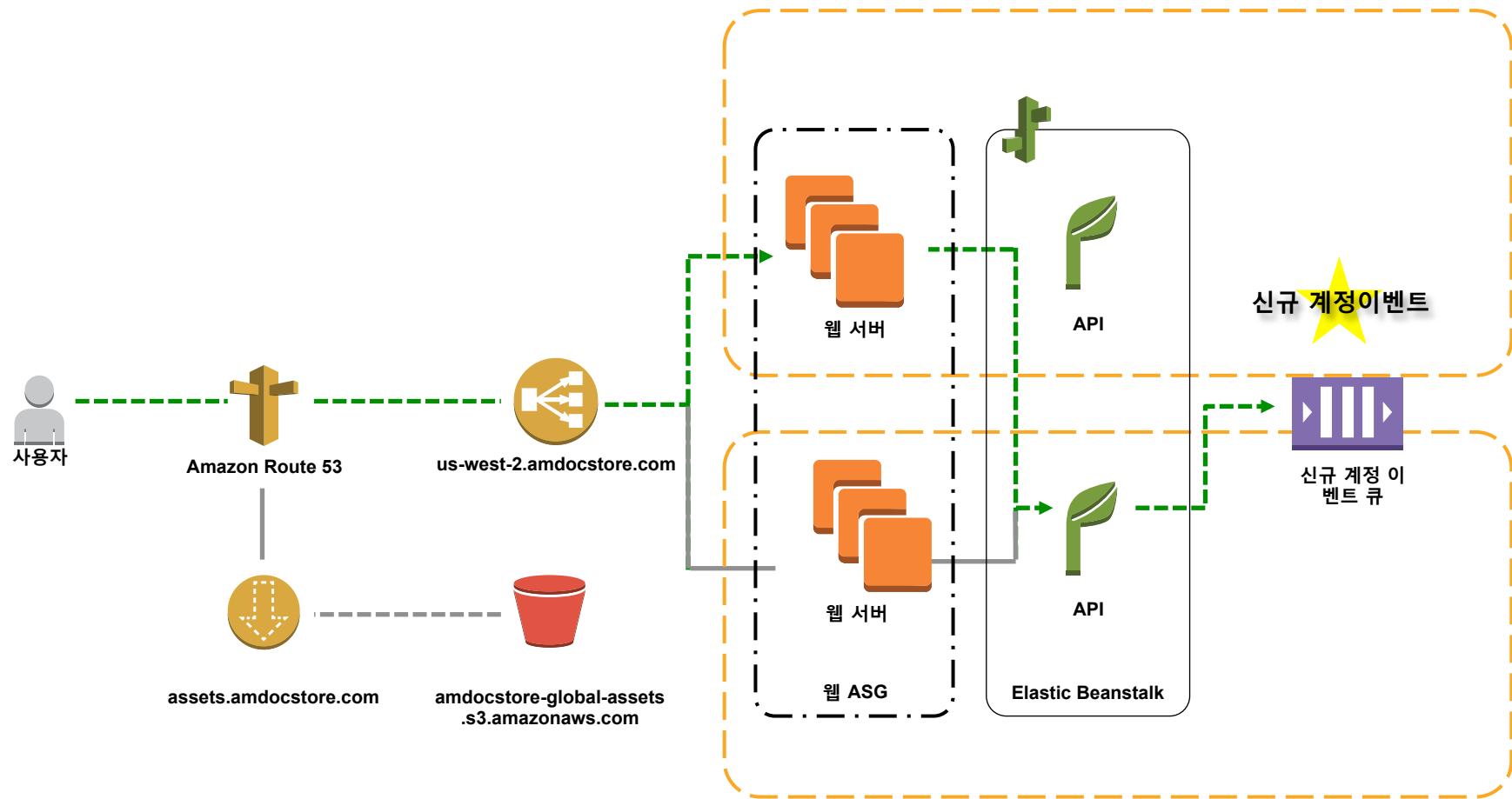
계정 생성: 본 애플리케이션의 다른 구성 요소에서 관심을 가질 수 있는 계정 가입 이벤트에 대해 살펴보겠습니다.

DocStore



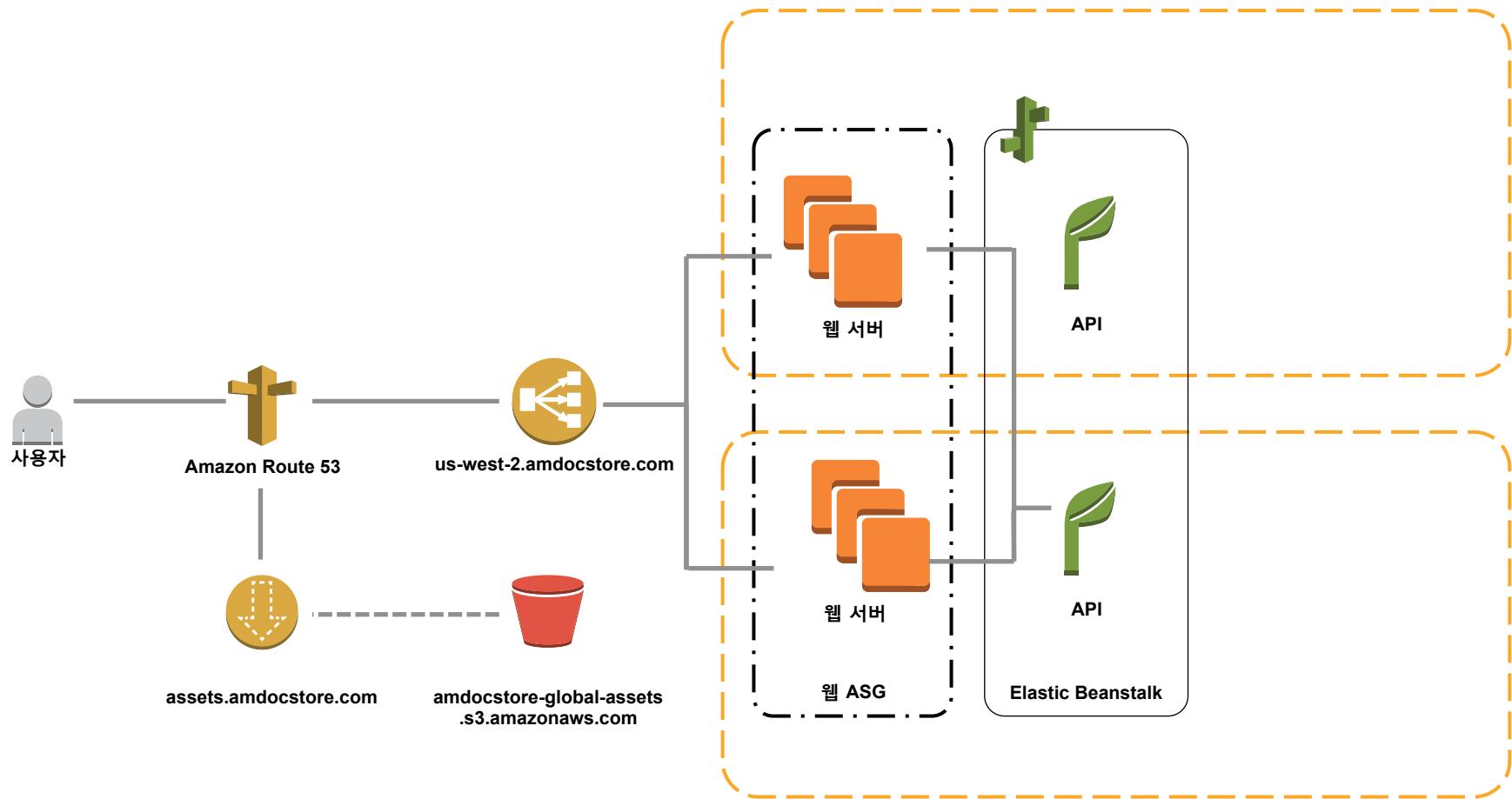
계정 생성: 이벤트에 관심이 있는 리스너, 즉 지역 간 계정 복제를 처리하는 서비스는 메시지 큐를 통해 이벤트를 수신합니다.

DocStore



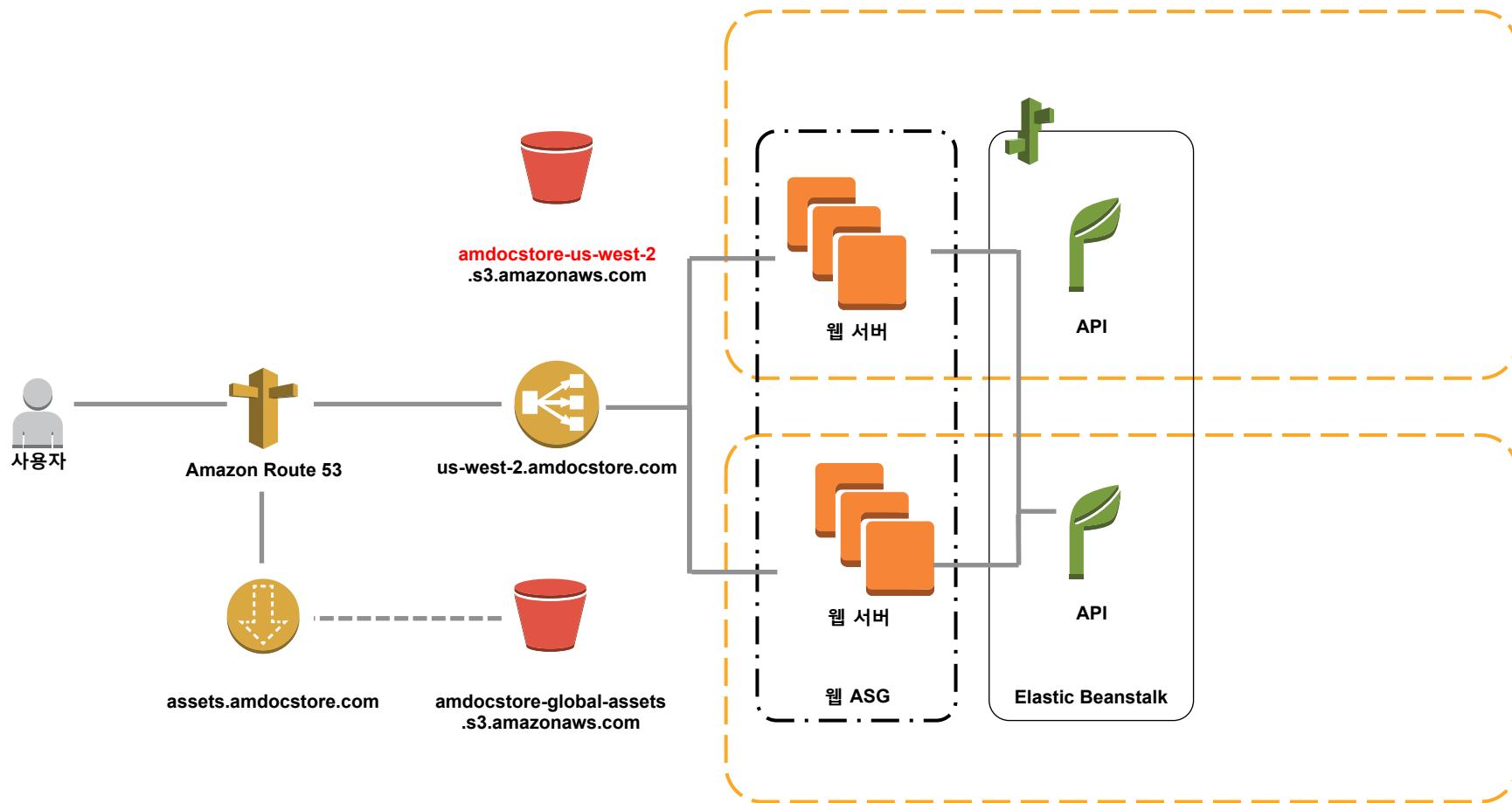
사용자가 업로드한 문서는 S3 버킷에 저장됩니다.

DocStore



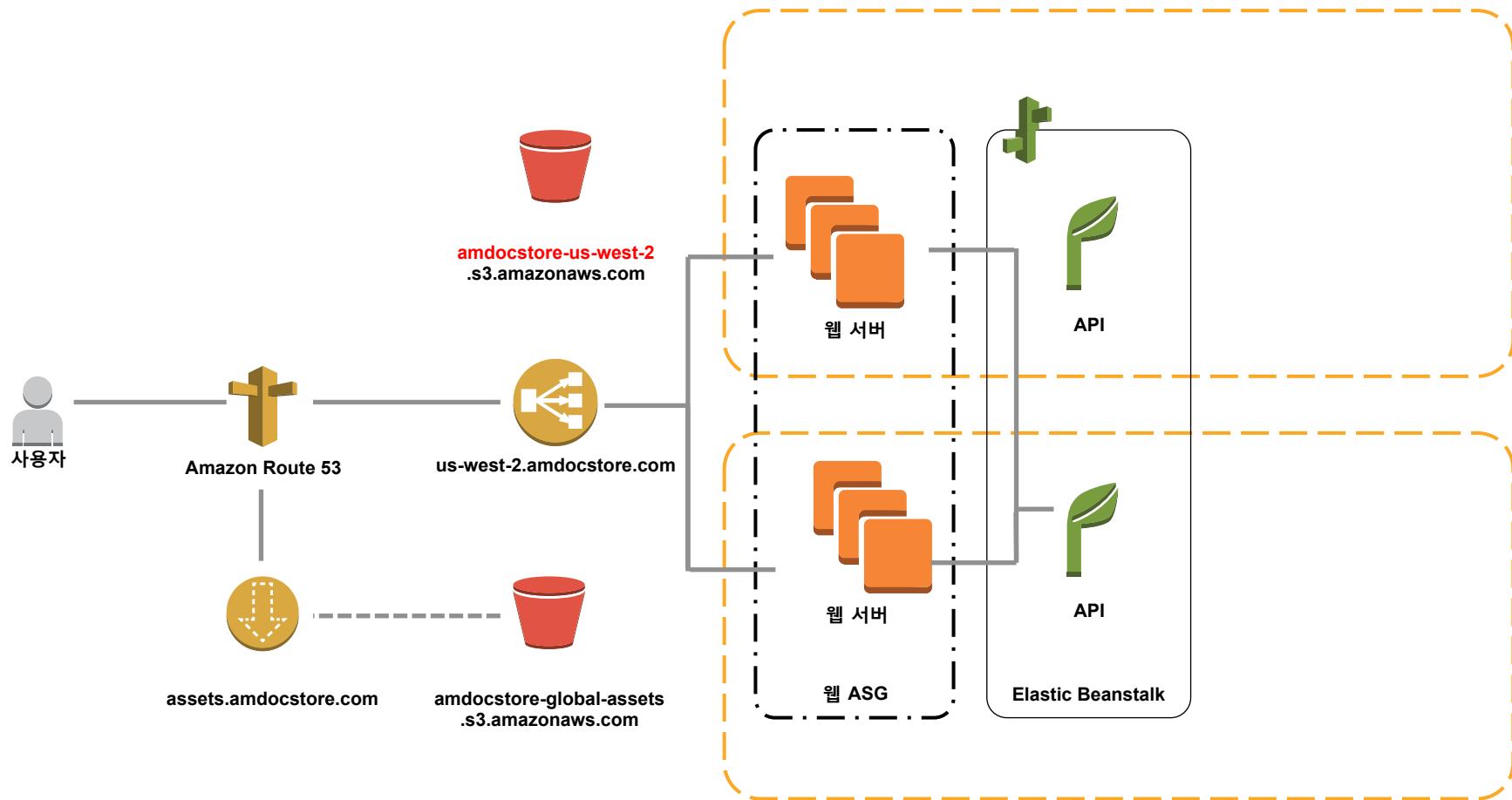
S3 버킷 이름에는 해당 버킷이 생성된 지역의 이름이 포함됩니다.

DocStore



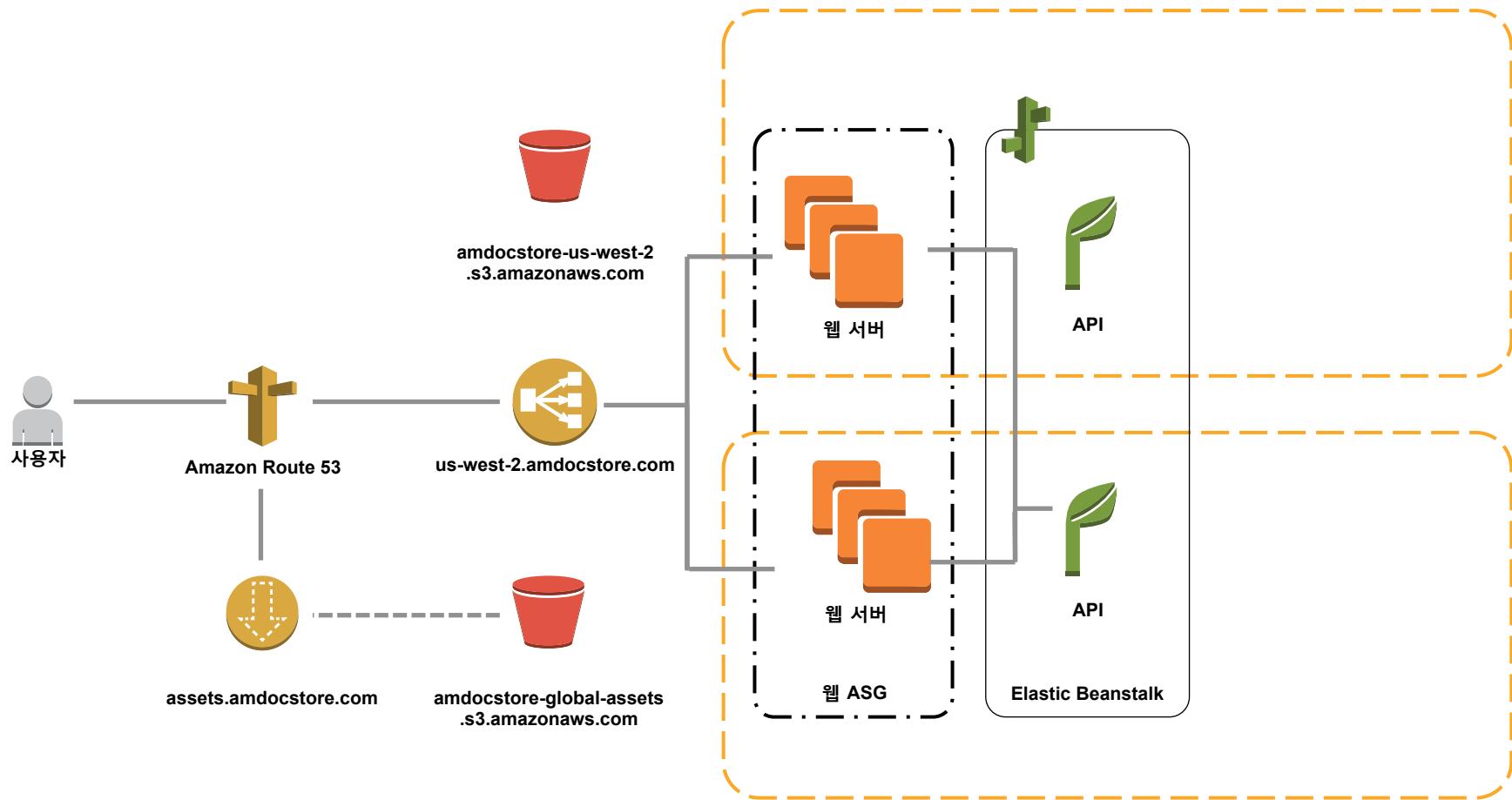
S3 버킷 이름에는 해당 버킷이 생성된 지역의 이름이 포함됩니다. 사용자는 해당 사용자의 홈 지역에 위치한 버킷으로 문서를 업로드합니다.

DocStore



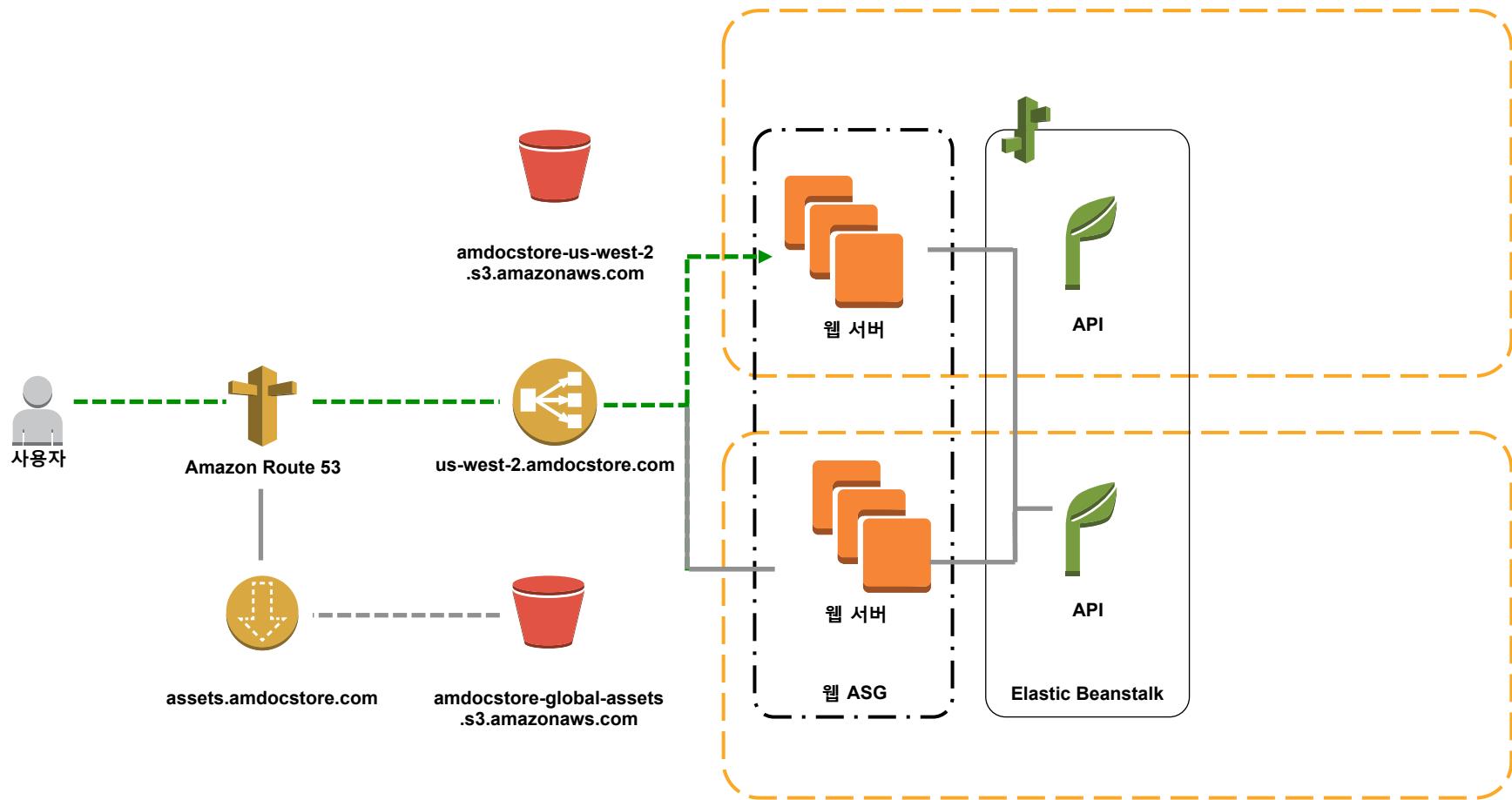
사용자가 PDF 문서를 업로드하려고 할 때 어떻게 처리되는지
살펴보겠습니다.

DocStore



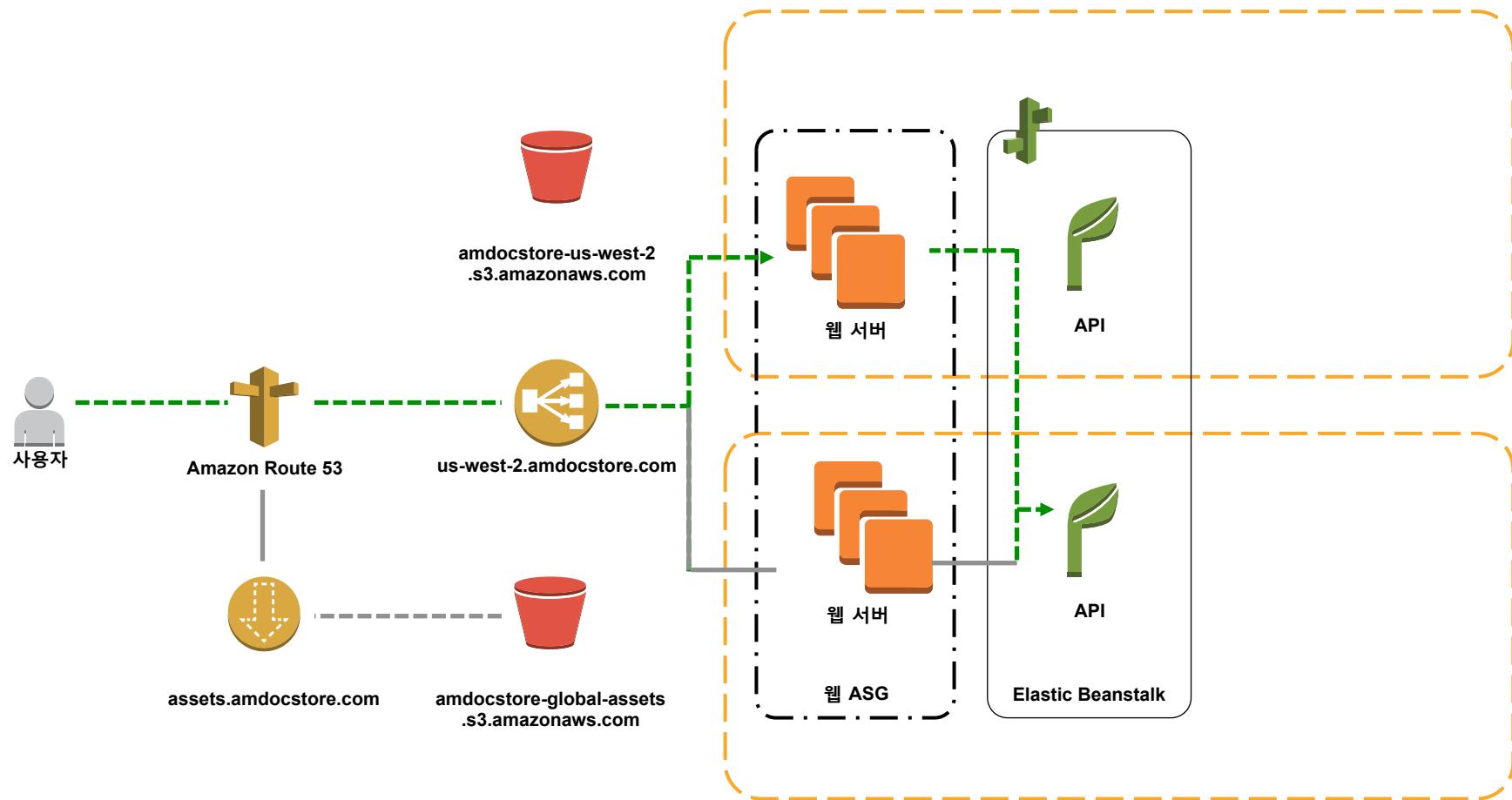
문서 업로드: 인증된 사용자는 웹 서버로부터 문서 업로드 폼을
이지를 요청합니다.

DocStore



문서 업로드: 웹 서버는 API로부터 서명된 HTML 양식을 요청하여 이를 사용자의 브라우저로 전달합니다.

DocStore



문서 업로드: 서명된 양식은 사용자의 브라우저로 전달됩니다.

DocStore

Upload New Doc

C:\Users\jon\docs\contract.pdf

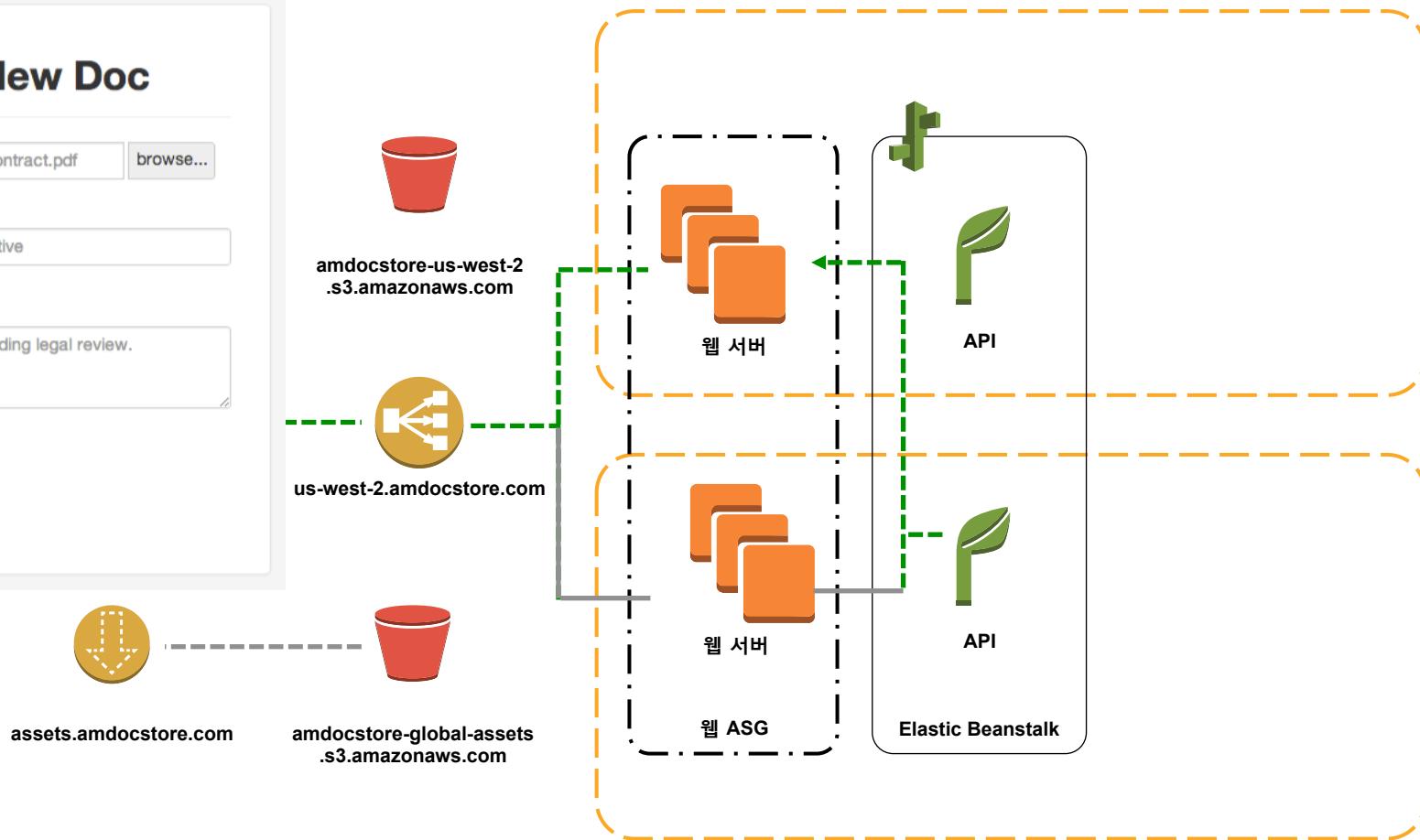
Tags

legal, contract, sensitive

Description

Client contracts. Pending legal review.

Make Public

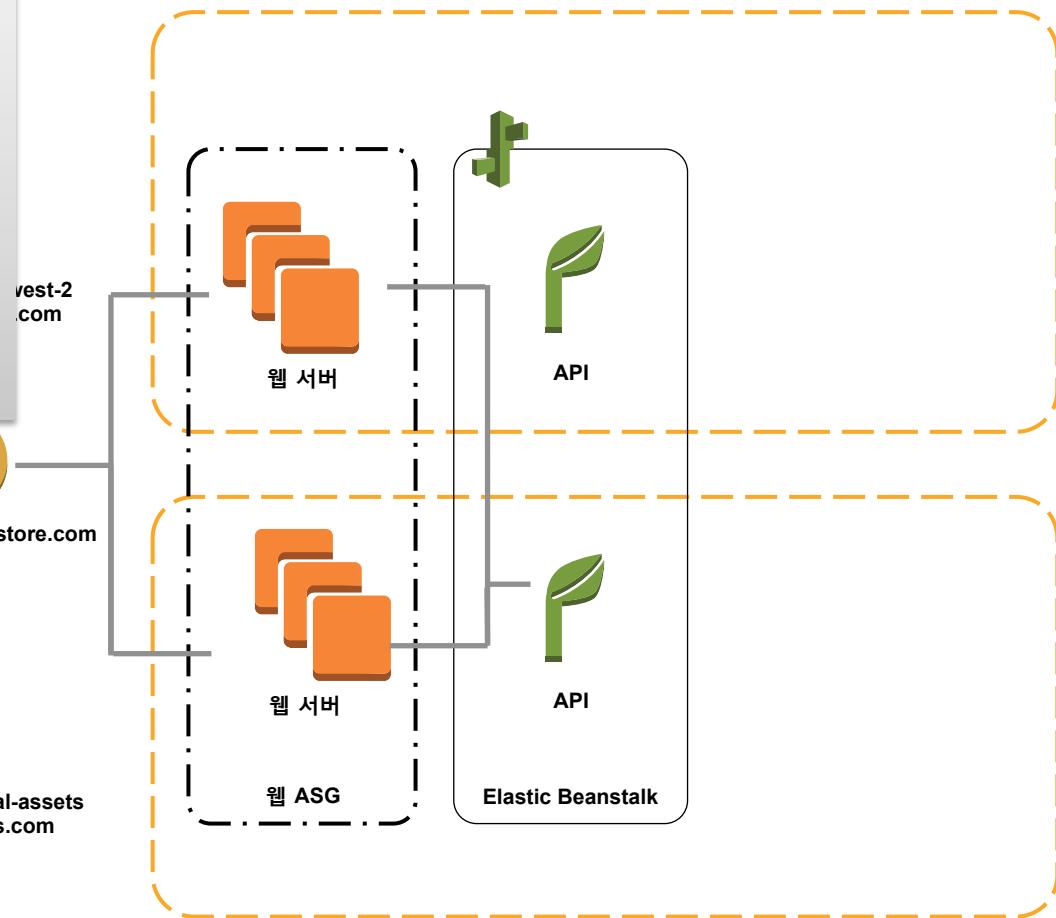
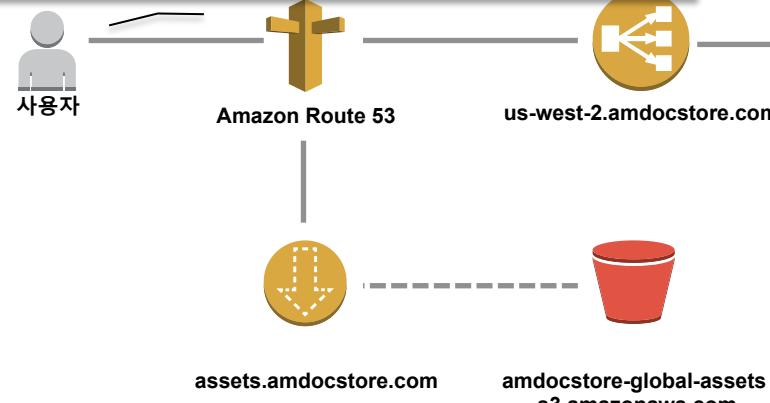


문서 업로드: 서명된 양식은 사용자가 문서를 S3에 직접 업로드할 수 있도록 허용합니다.

DocStore

```
<form action="amdocstore-us-west-2...">
  <!-- 숨겨진 입력 필드가 제약을 시행 -->
  <input name="expiration" value="in 90 seconds" />
  <input name="key" value="0dfa-238h-2j8a-92cg" />
  <input name="signature" value="AS290a88aSe..." />

  <!-- 텍스트 입력이 S3 내 객체 메타데이터가 됨 -->
  <input name="x-amz-meta-tags" type="text" />
  <input name="x-amz-meta-description" type="text" />
</form>
```

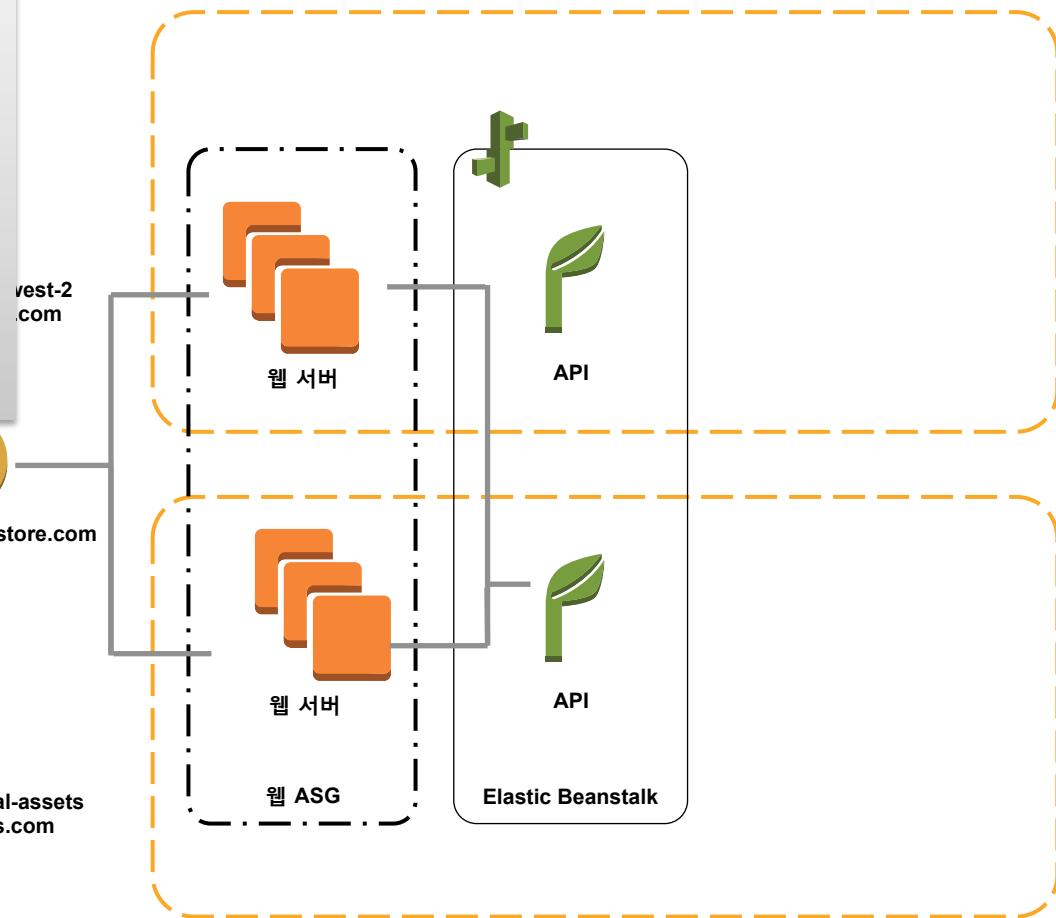
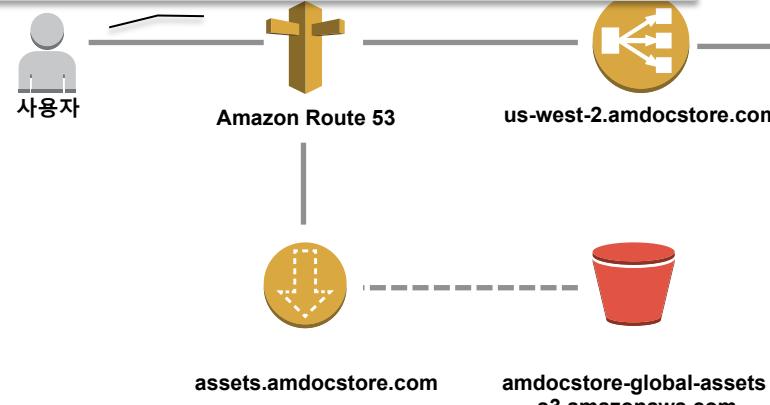


문서 업로드: 양식은 사용자가 업로드할 수 있도록 해주는 인증 정보를 숨은 입력으로 포함합니다.

DocStore

```
<form action="amdocstore-us-west-2...">
  <!-- 숨겨진 입력 필드가 제약을 시행-->
  <input name="expiration" value="in 90 seconds" />
  <input name="key" value="0dfa-238h-2j8a-92cg" />
  <input name="signature" value="AS290a88aSe..." />

  <!--텍스트 입력이 S3 내 객체 메타데이터가 됨-->
  <input name="x-amz-meta-tags" type="text" />
  <input name="x-amz-meta-description" type="text" />
</form>
```

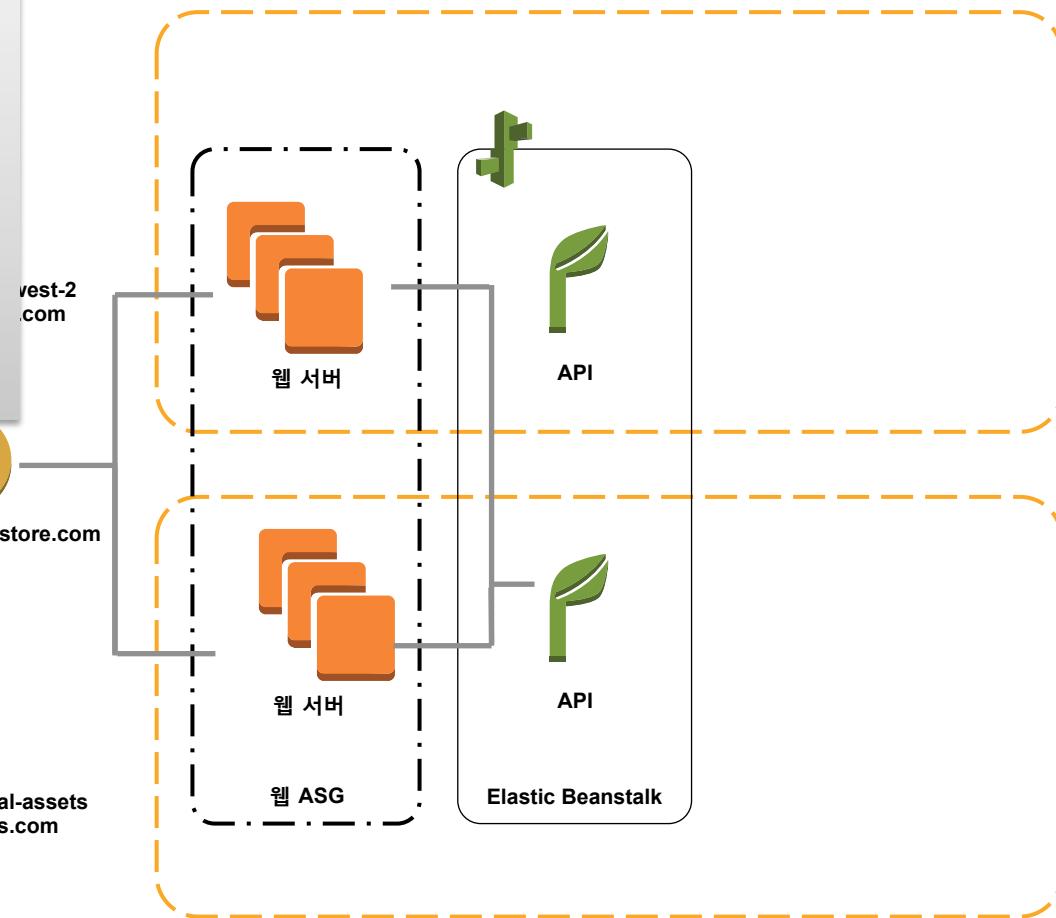
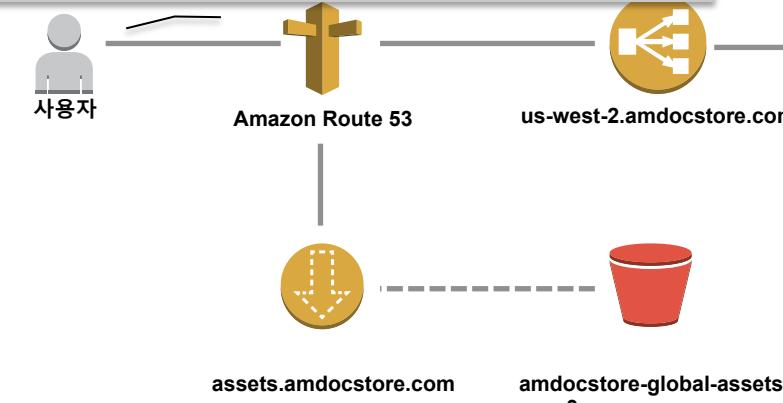


문서 업로드: 양식은 사용자가 업로드할 수 있도록 해주는 인증 정보를 숨은 입력으로 포함합니다. 만료 기한은 양식이 얼마나 오래 유효한지를 나타냅니다.

DocStore

```
<form action="amdocstore-us-west-2...">
  <!-- 숨겨진 입력 필드가 제약을 시행-->
  <input name="expiration" value="in 90 seconds" />
  <input name="key" value="0dfa-238h-2j8a-92cg" />
  <input name="signature" value="AS290a88aSe..." />

  <!--텍스트 입력이 S3 내 객체 메타데이터가 됨-->
  <input name="x-amz-meta-tags" type="text" />
  <input name="x-amz-meta-description" type="text" />
</form>
```

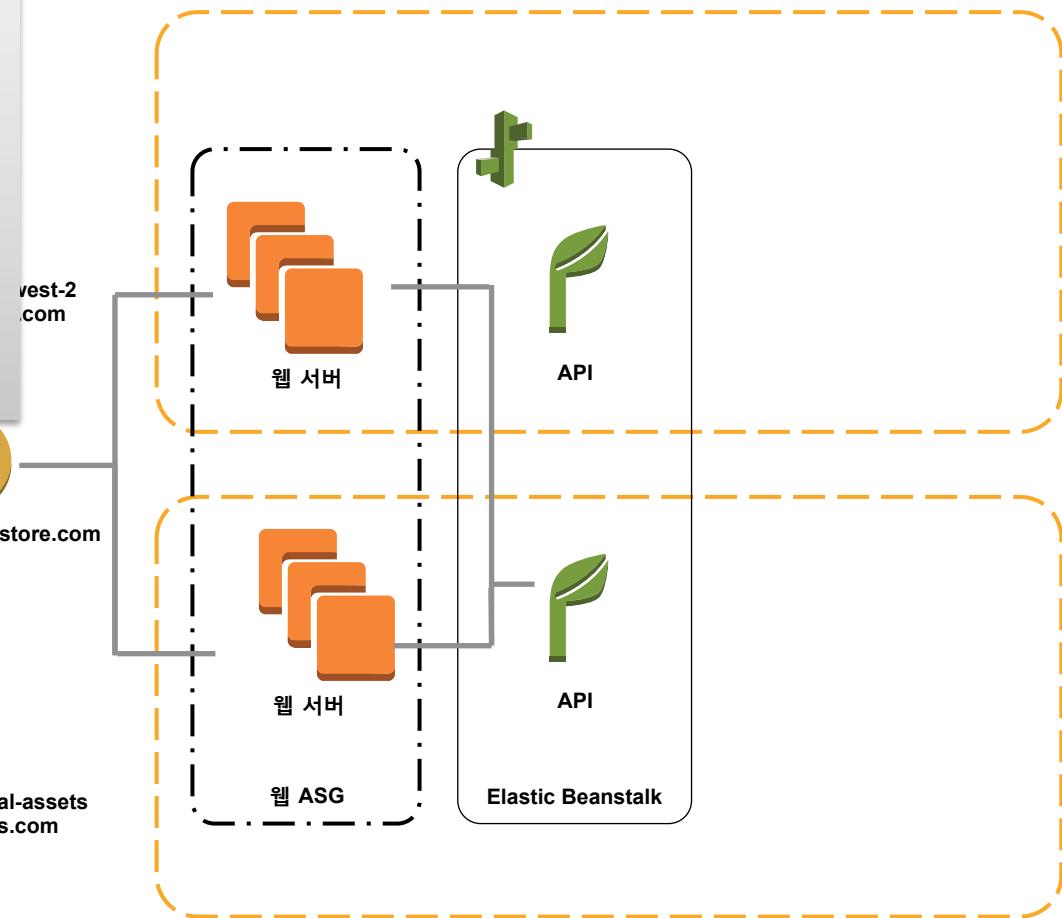
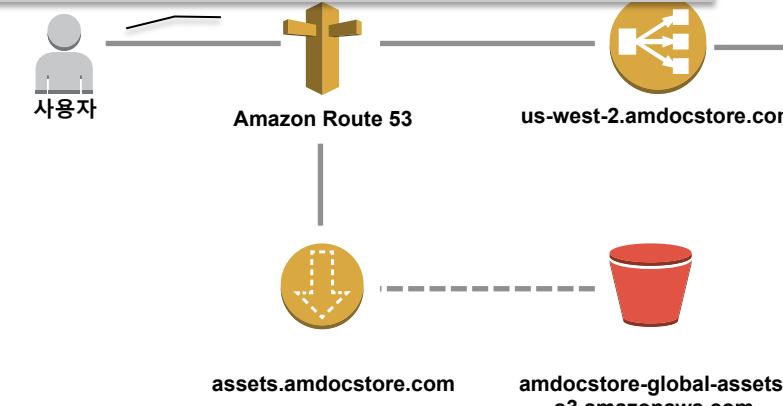


문서 업로드: 양식은 사용자가 업로드할 수 있도록 해주는 인증 정보를 숨은 입력으로 포함합니다. 키는 S3 내에서 객체 이름을 지정하도록 허용하여 이름 충돌을 방지합니다.

DocStore

```
<form action="amdocstore-us-west-2...">
  <!-- 숨겨진 입력 필드가 제약을 시행-->
  <input name="expiration" value="in 90 seconds" />
  <input name="key" value="0dfa-238h-2j8a-92cg" />
  <input name="signature" value="AS290a88aSe..." />

  <!--텍스트 입력이 S3 내 객체 메타데이터가 됨-->
  <input name="x-amz-meta-tags" type="text" />
  <input name="x-amz-meta-description" type="text" />
</form>
```

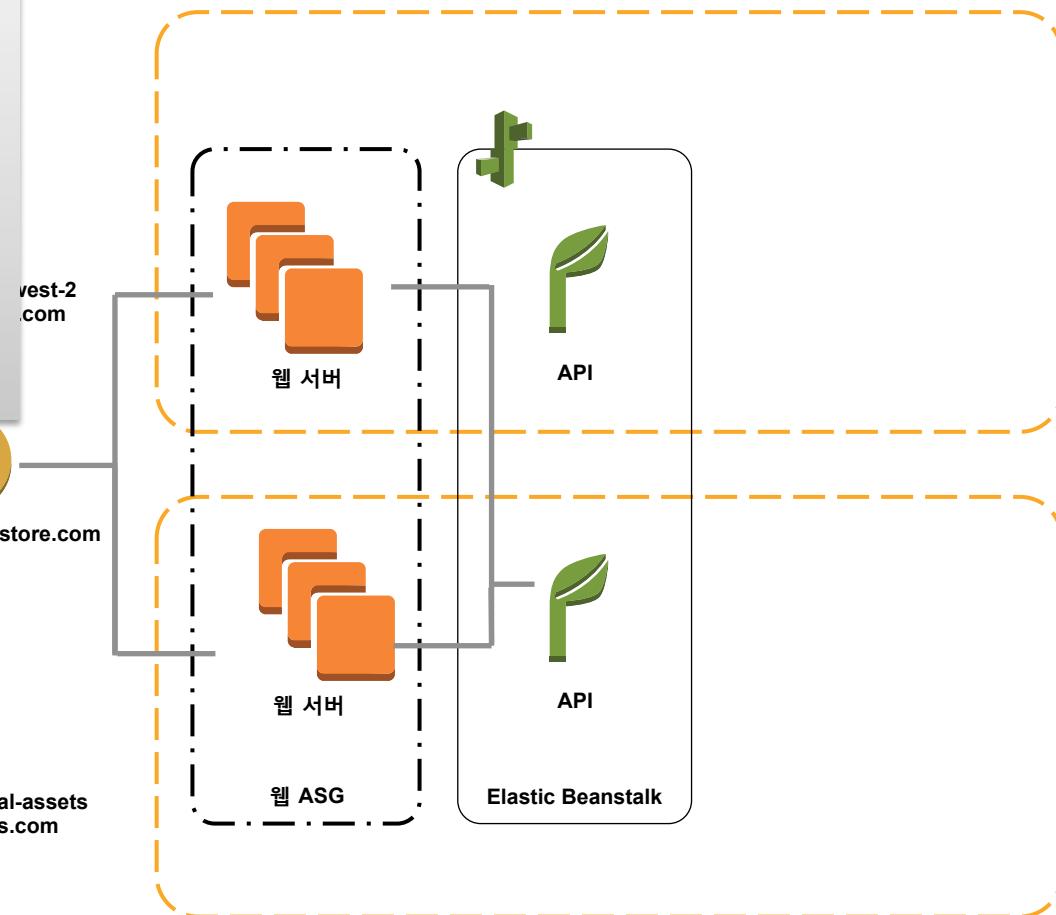
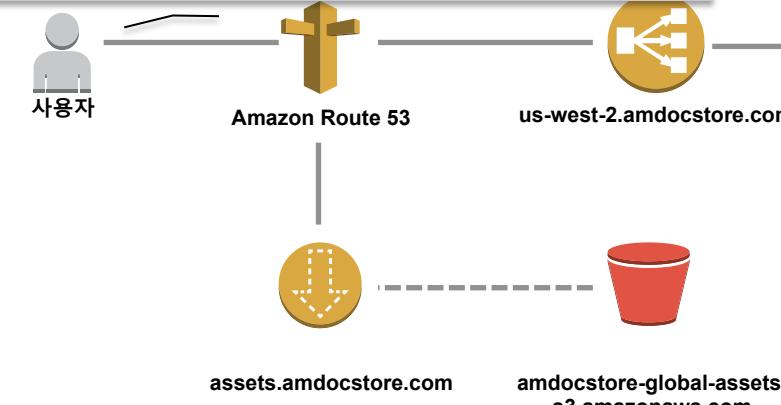


문서 업로드: 양식은 사용자가 업로드할 수 있도록 해주는 인증 정보를 숨은 입력으로 포함합니다. 서명은 양식이 변조되었을 경우(예: 사용자가 만료 기한을 변경) 업로드를 거부할 수 있도록 합니다.

DocStore

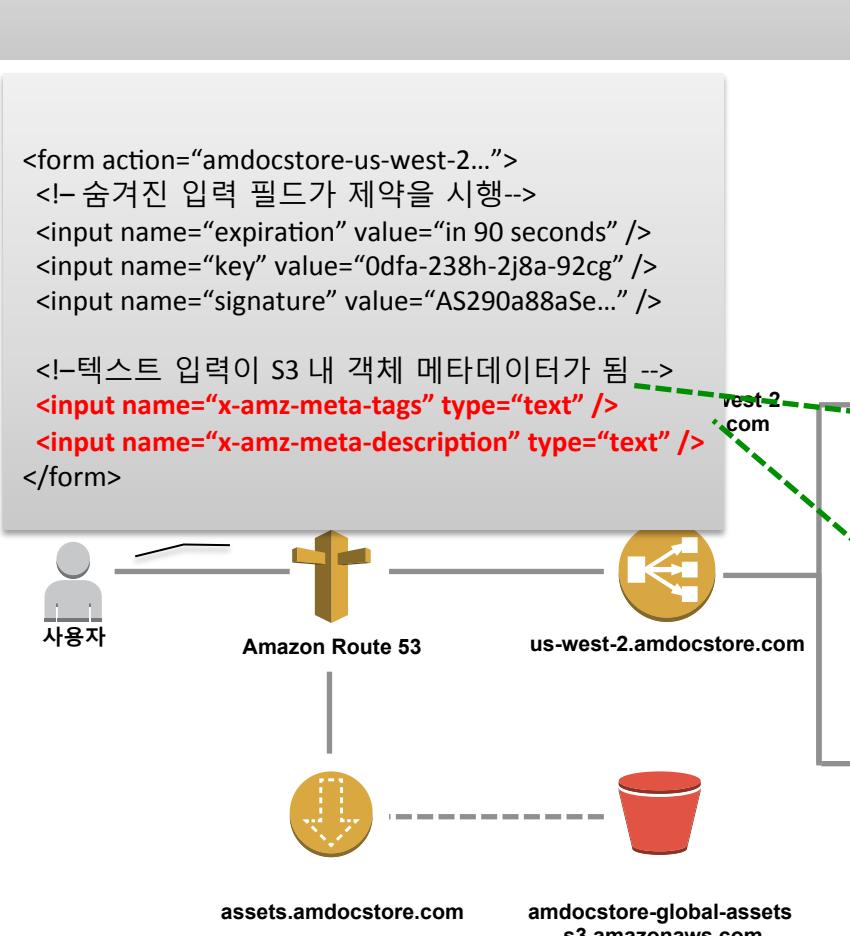
```
<form action="amdocstore-us-west-2...">
  <!-- 숨겨진 입력 필드가 제약을 시행-->
  <input name="expiration" value="in 90 seconds" />
  <input name="key" value="0dfa-238h-2j8a-92cg" />
  <input name="signature" value="AS290a88aSe..." />

  <!--텍스트 입력이 S3 내 객체 메타데이터가 됨-->
  <input name="x-amz-meta-tags" type="text" />
  <input name="x-amz-meta-description" type="text" />
</form>
```



문서 업로드: 사용자가 입력하는 x-amz-meta- 접두사를 갖는 텍스트 입력 필드는 업로드된 객체의 S3 객체 메타데이터로 등록됩니다.

DocStore



Upload New Doc

C:\Users\jon\docs\contract.pdf

Tags
legal, contract, sensitive

Description
Client contracts. Pending legal review.

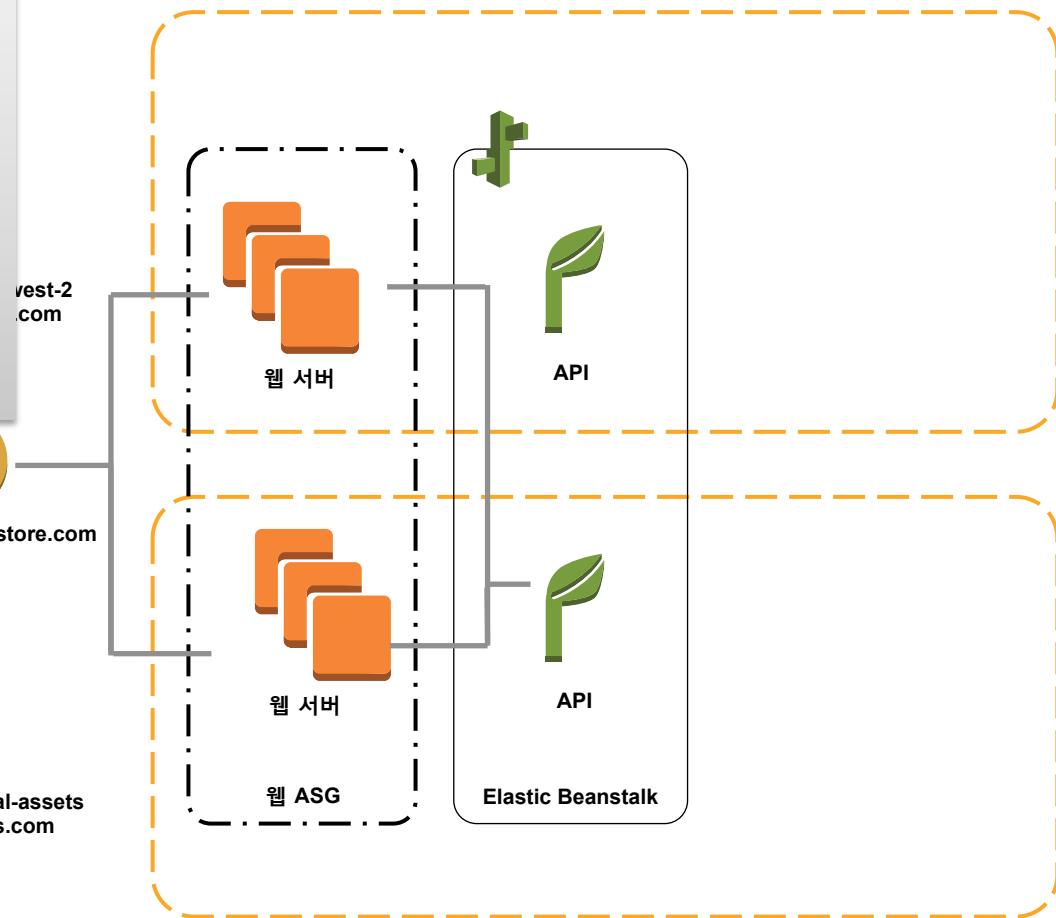
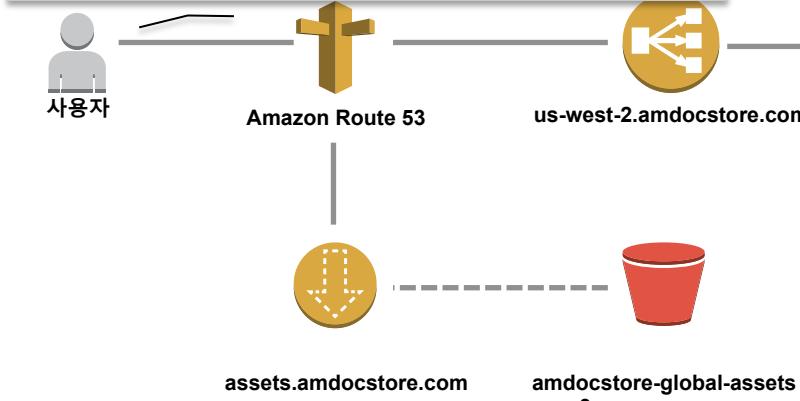
Make Public



문서 업로드: 마지막으로, API가 양식 생성 시 지정한 **success_action_redirect**는 S3가 업로드 완료 후 사용자 브라우저를 어디로 리디렉션할지 알려줍니다.

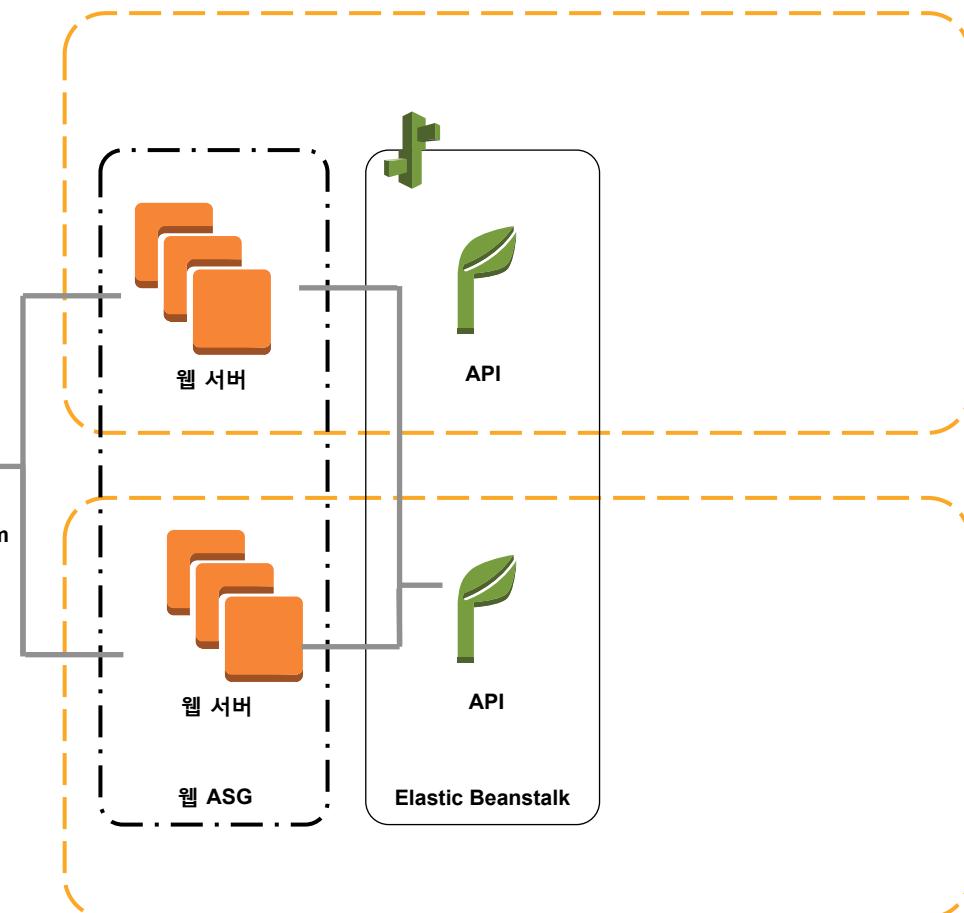
DocStore

```
<form action="amdocstore-us-west-2...">  
    <!-- 숨겨진 입력 필드가 제약을 시행 -->  
    <input name="success_action_redirect"  
        value="https://amdocstore.com/doc-uploaded" />  
</form>
```



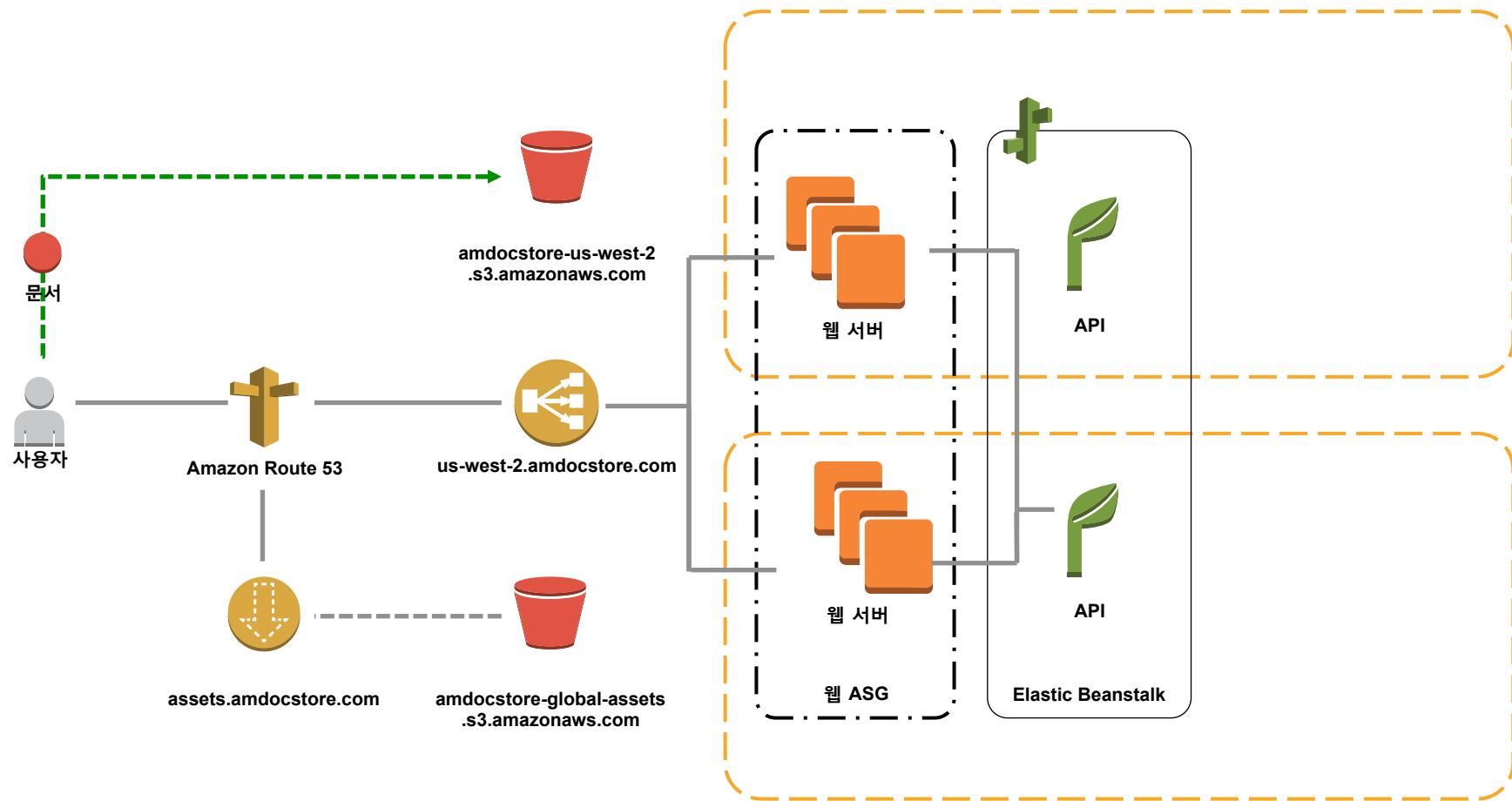
문서 업로드: 이벤트를 받아 처리할 웹 서버상의 특정 페이지로 사용자를 리디렉션합니다.

DocStore



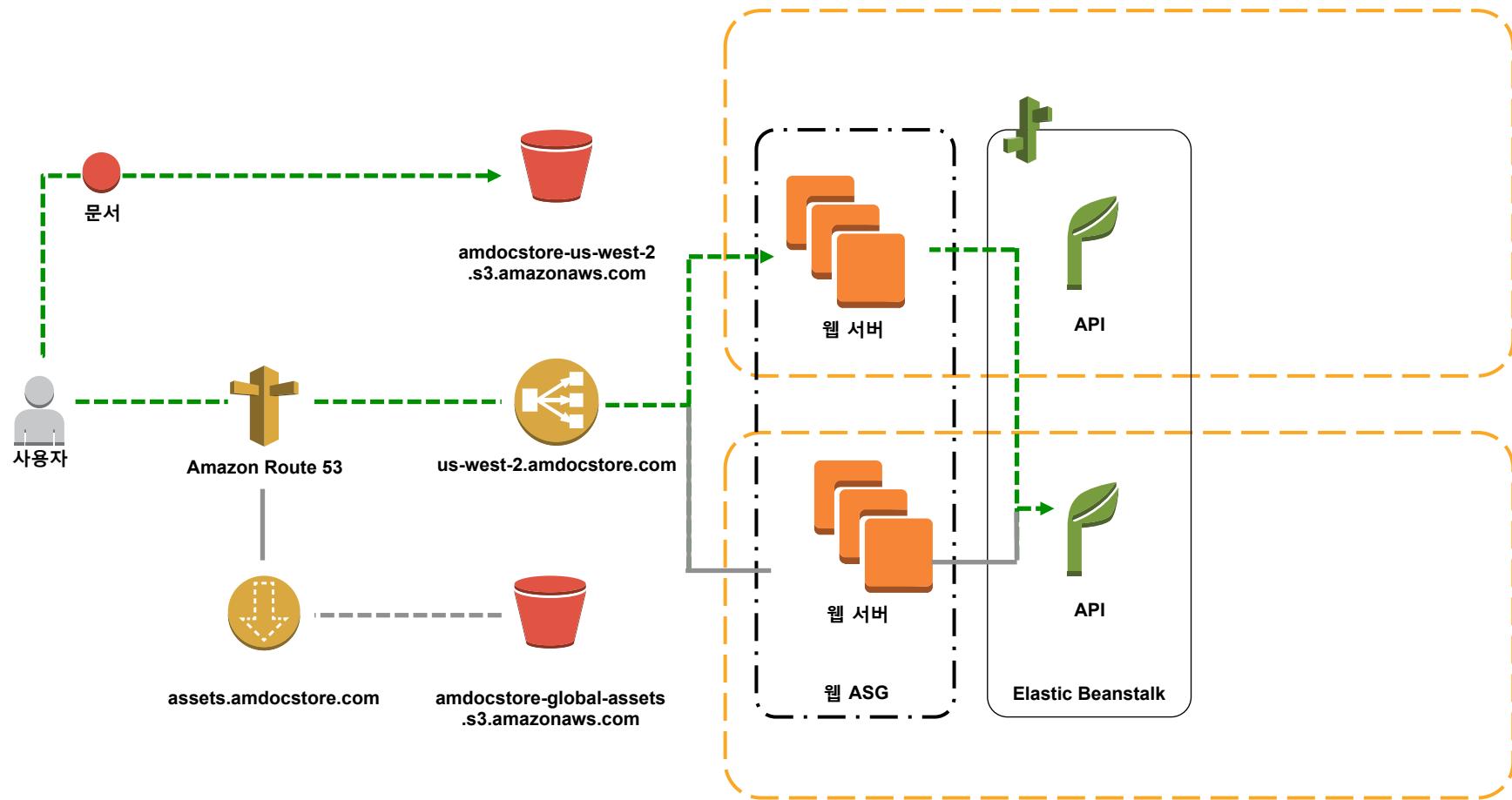
문서 업로드: S3로 문서가 업로드되는 동안.

DocStore



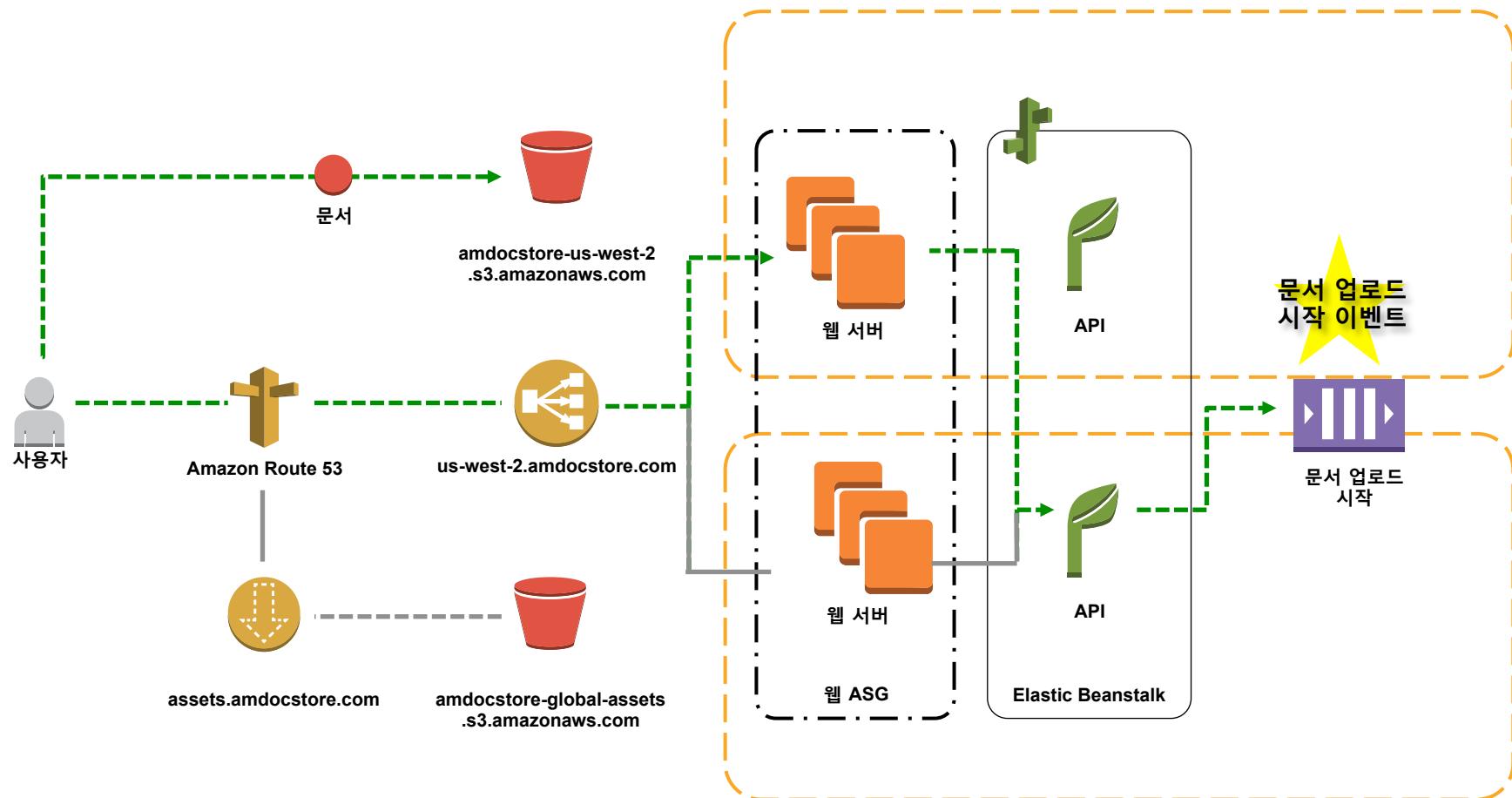
문서 업로드: S3로 문서가 업로드되는 동안 업로드가 시작되었음을 웹 서버에 비동기적으로 통지합니다.

DocStore



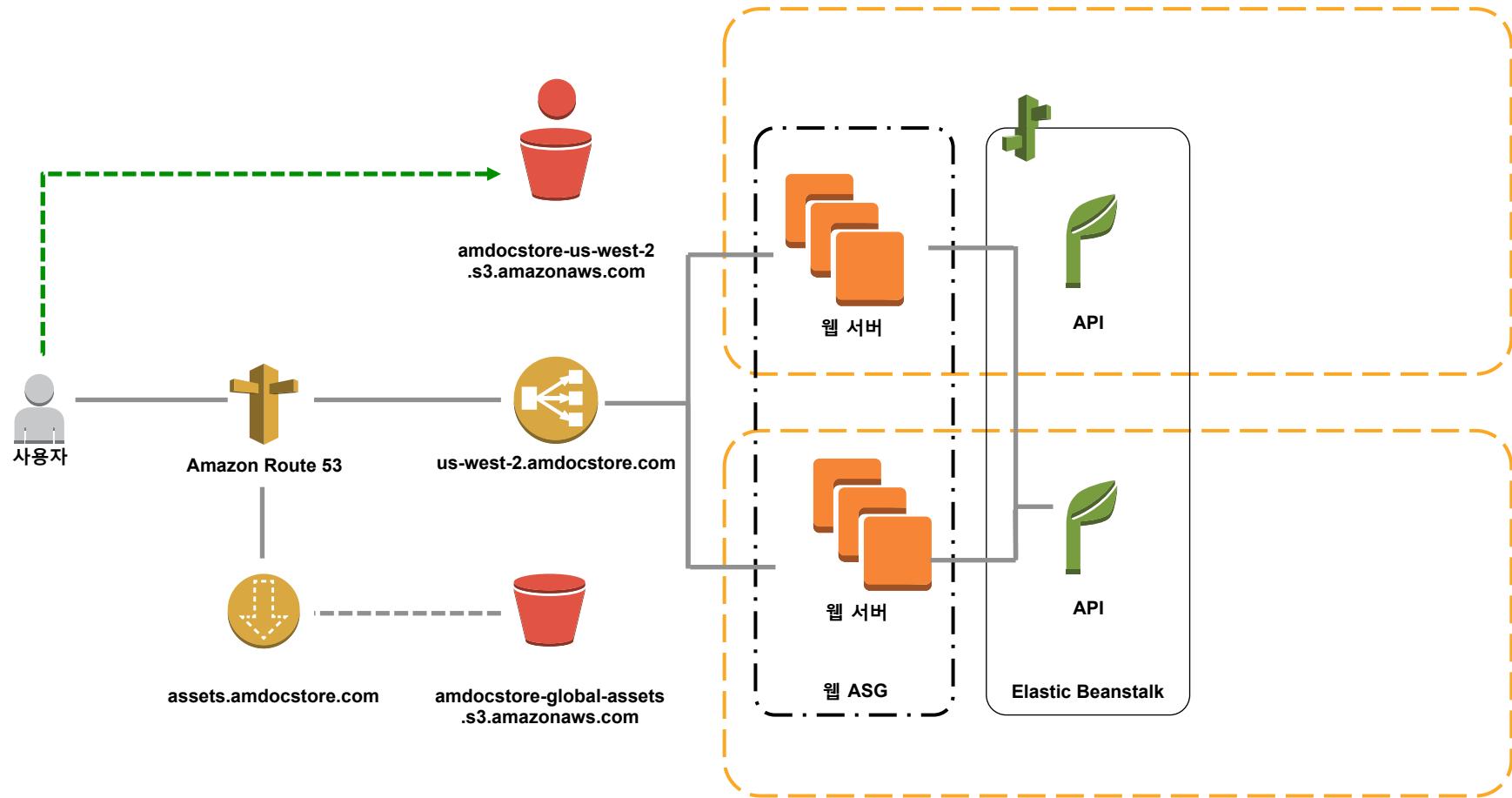
문서 업로드: API는 이벤트에 대한 정보를 담아 메시지를 큐에 넣습니다. 나중에 DocStore의 다른 구성 요소에서 해당 메시지를 처리합니다.

DocStore



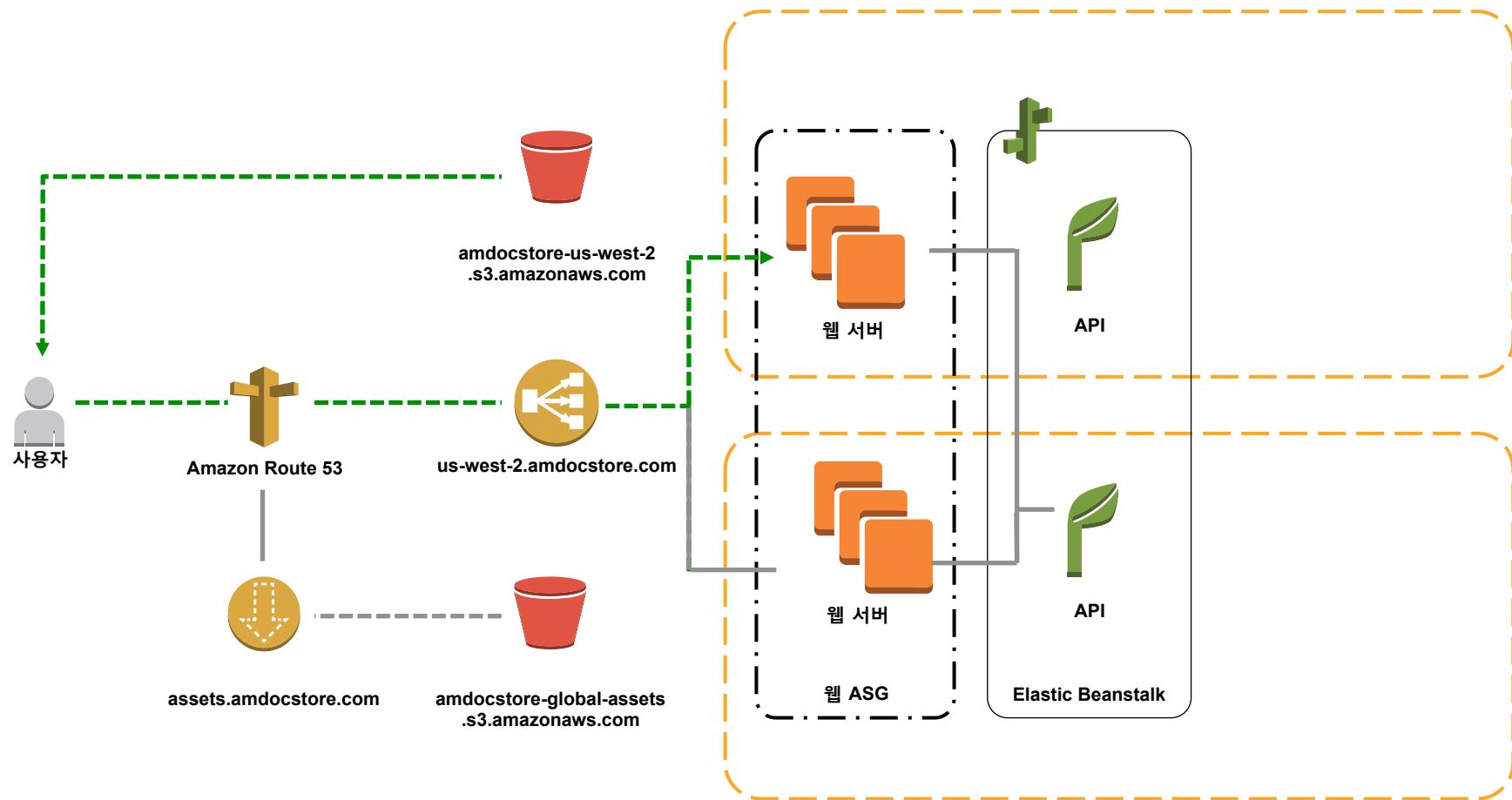
문서 업로드: 업로드가 완료되면.

DocStore



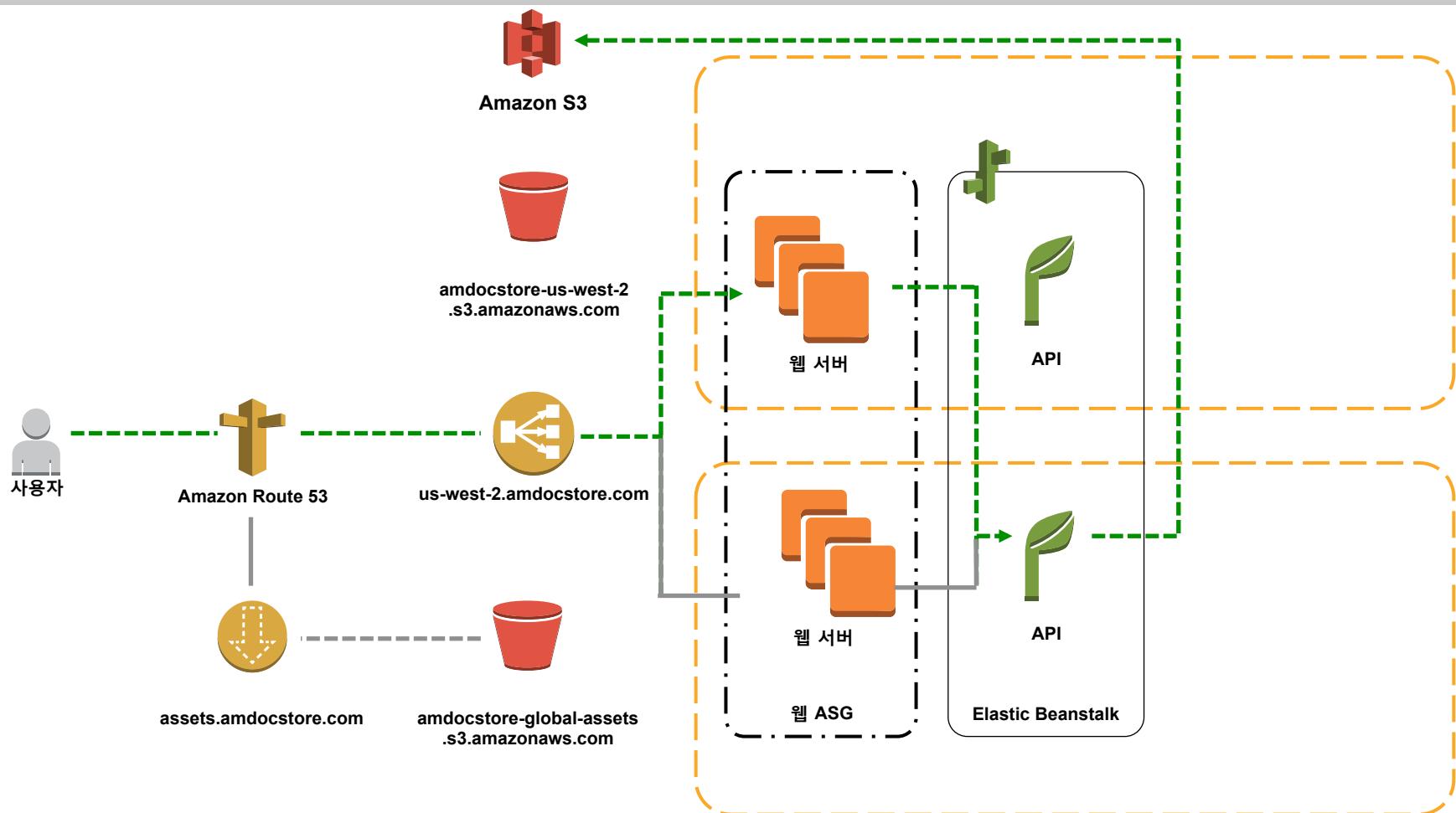
문서 업로드: 업로드가 완료되면 S3는 사용자를 이전에 지정된 URL로 리디렉션합니다.

DocStore



문서 업로드: API는 크기, 유형, 사용자 지정 정보와 같은 객체에 대한 메타데이터를 S3 API를 통해 확인합니다.

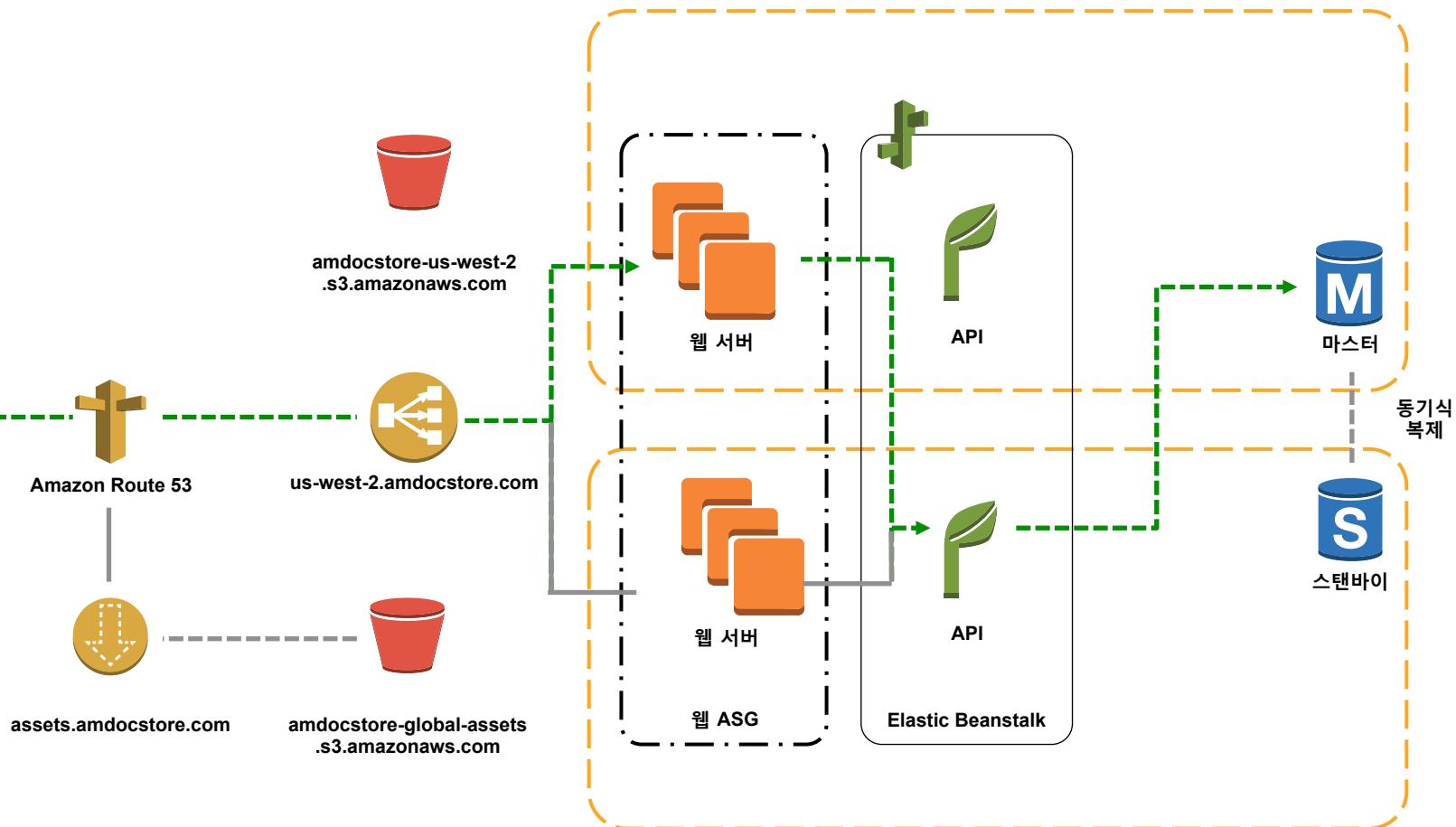
DocStore



문서 업로드: 객체 메타데이터는 고가용성을 위해
다중 AZ 모드로 구동되고 있는 **RDS** 인스턴스에
저장됩니다.

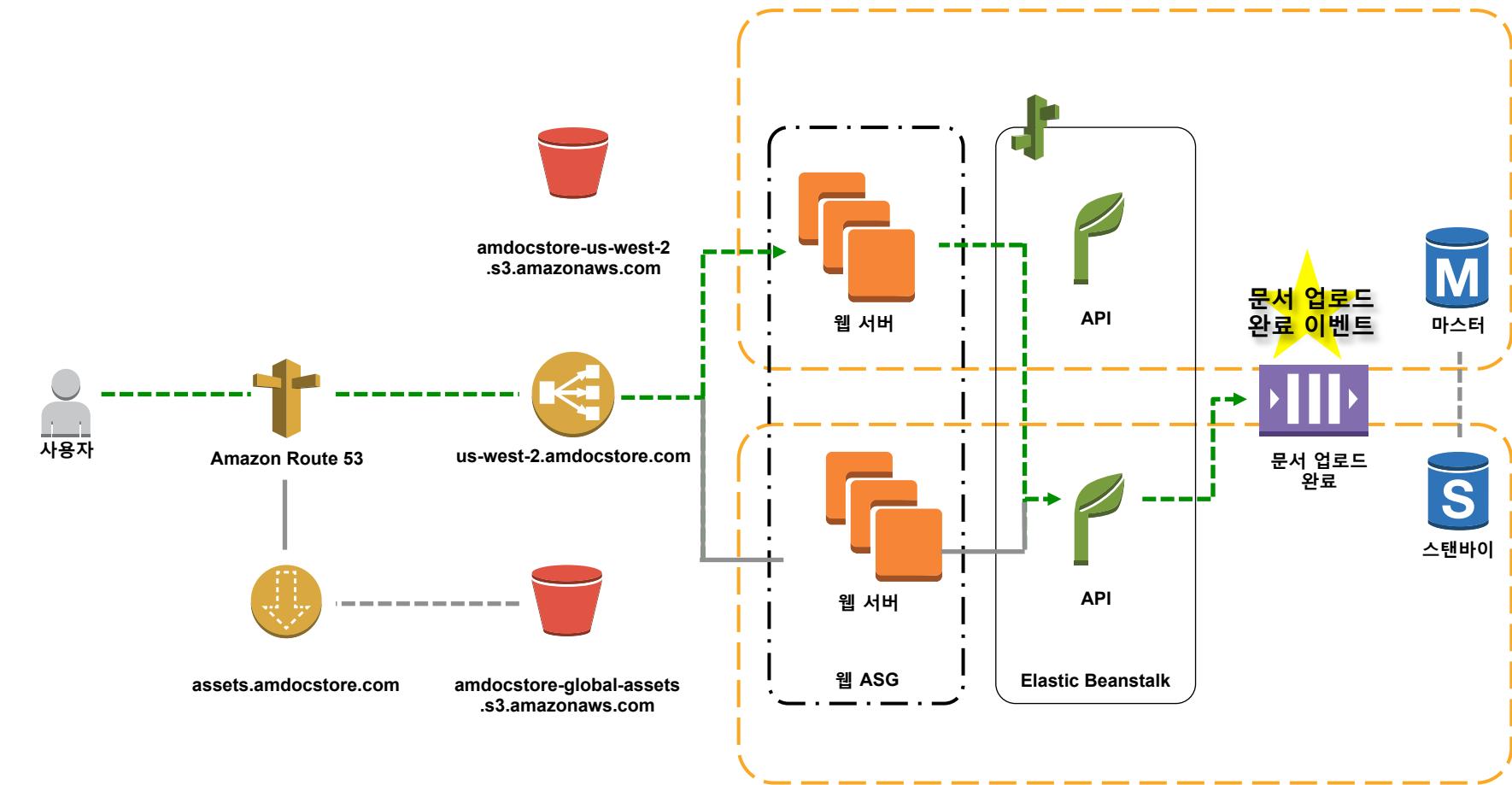


DocStore



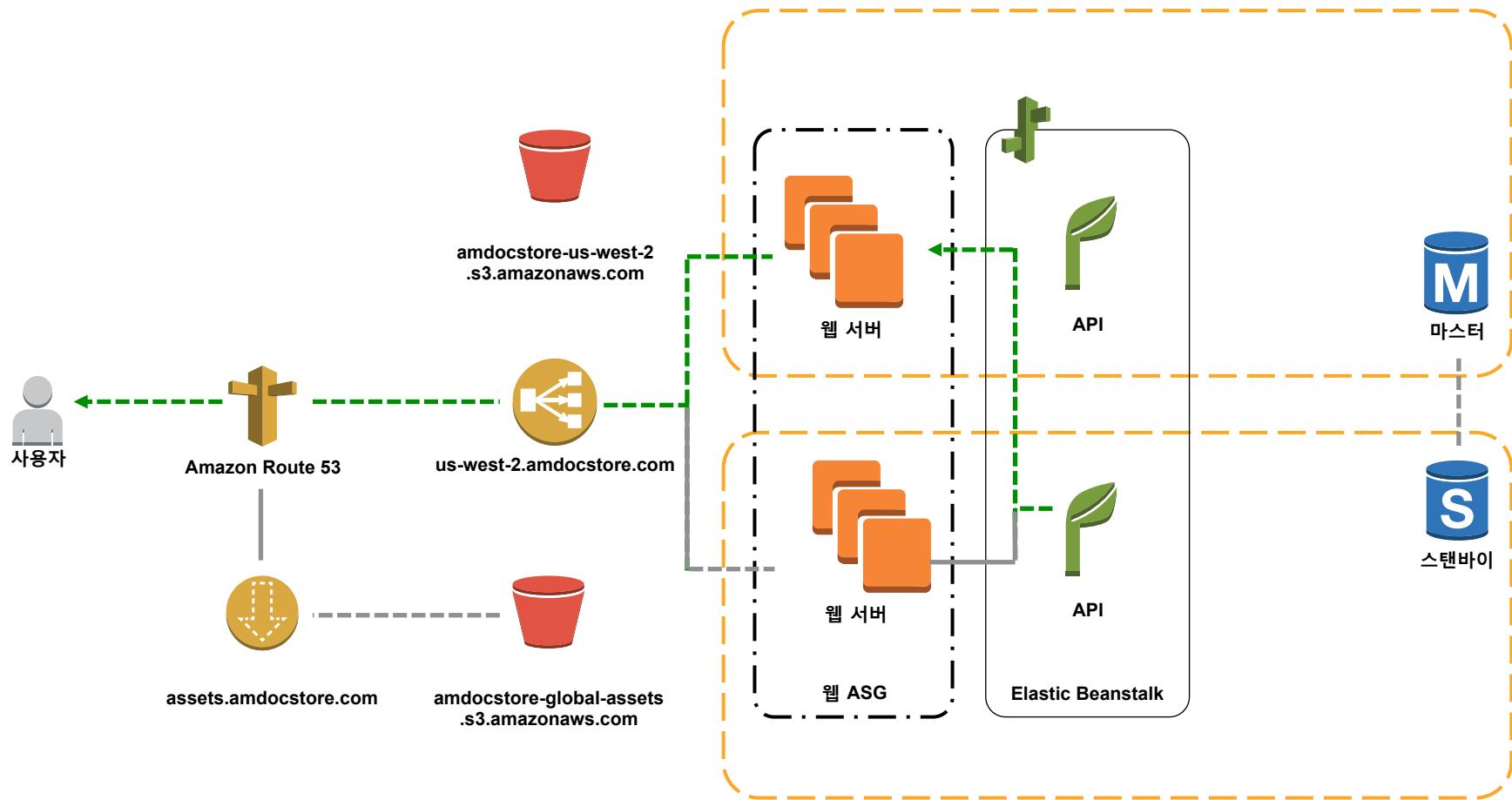
문서 업로드: 마지막으로, API는 문서 업로드가 완료되었음을 표시하고 업로드된 객체의 S3 식별자를 포함하는 메시지를 큐에 넣습니다.

DocStore



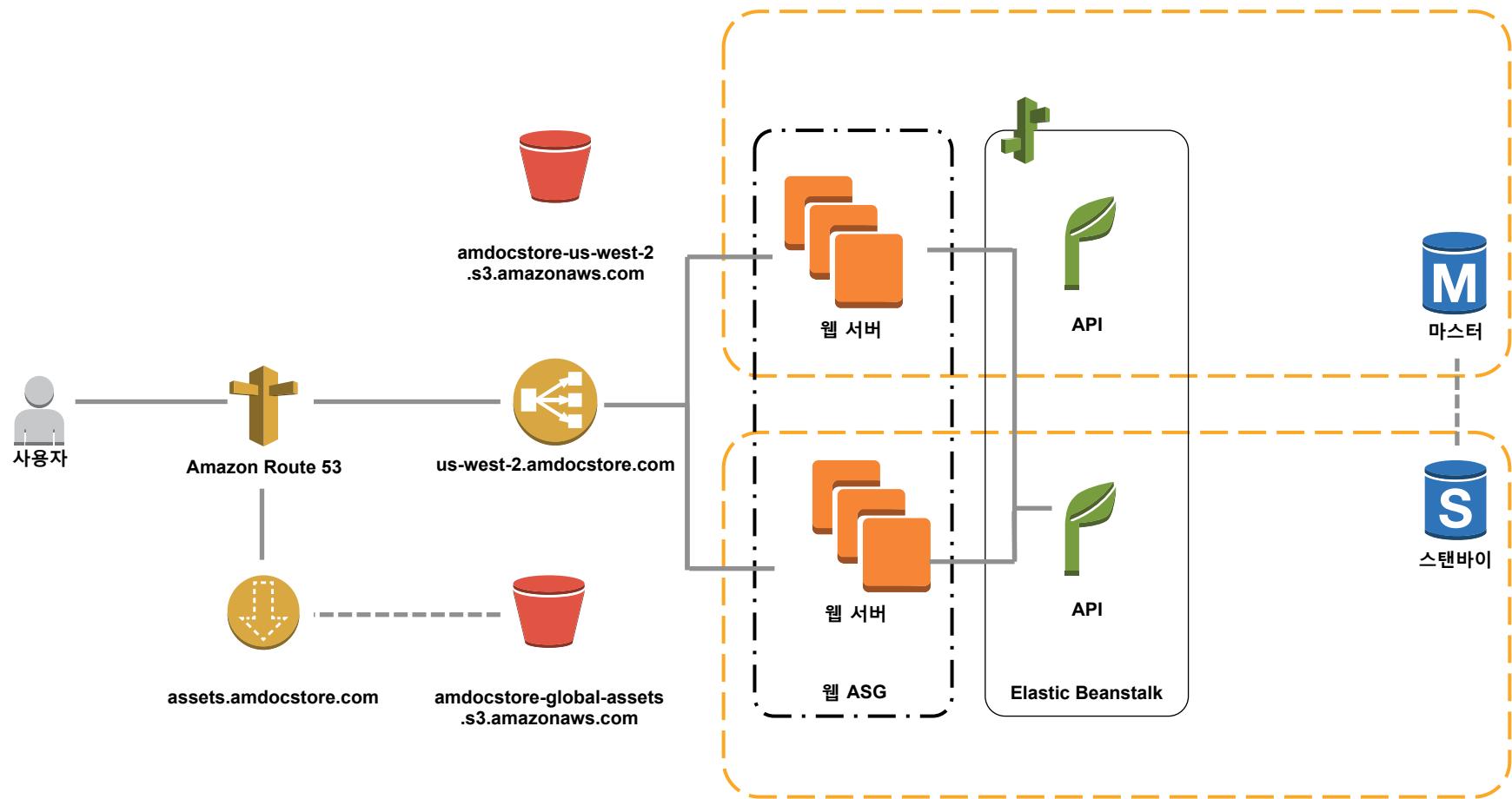
문서 업로드: 사용자는 자신의 문서 목록 페이지로 리디렉션 됩니다.

DocStore



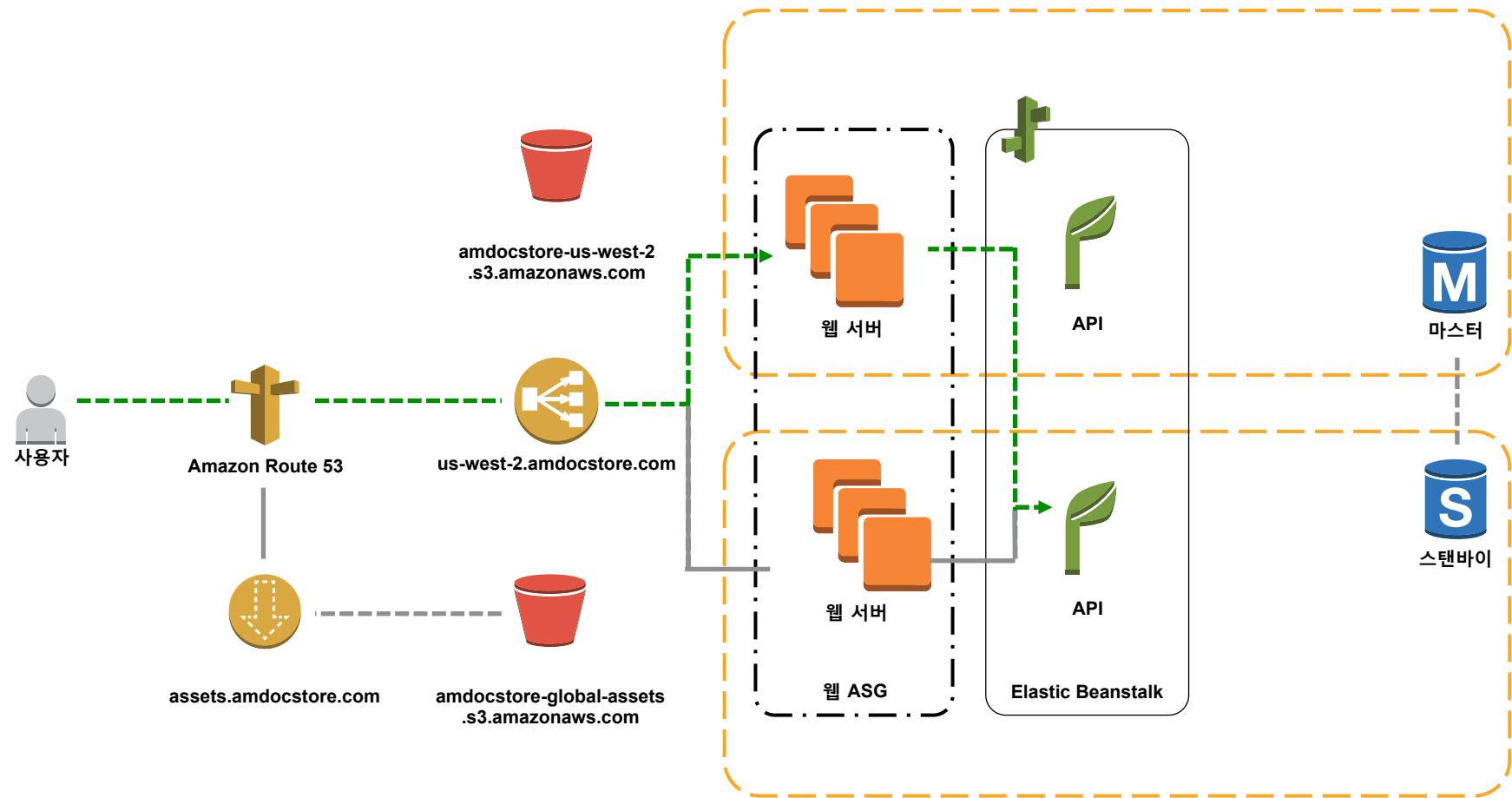
사용자가 자신의 문서를 조회하고 다운로드하려고 할 때 어떻게 처리되는지 살펴보겠습니다.

DocStore



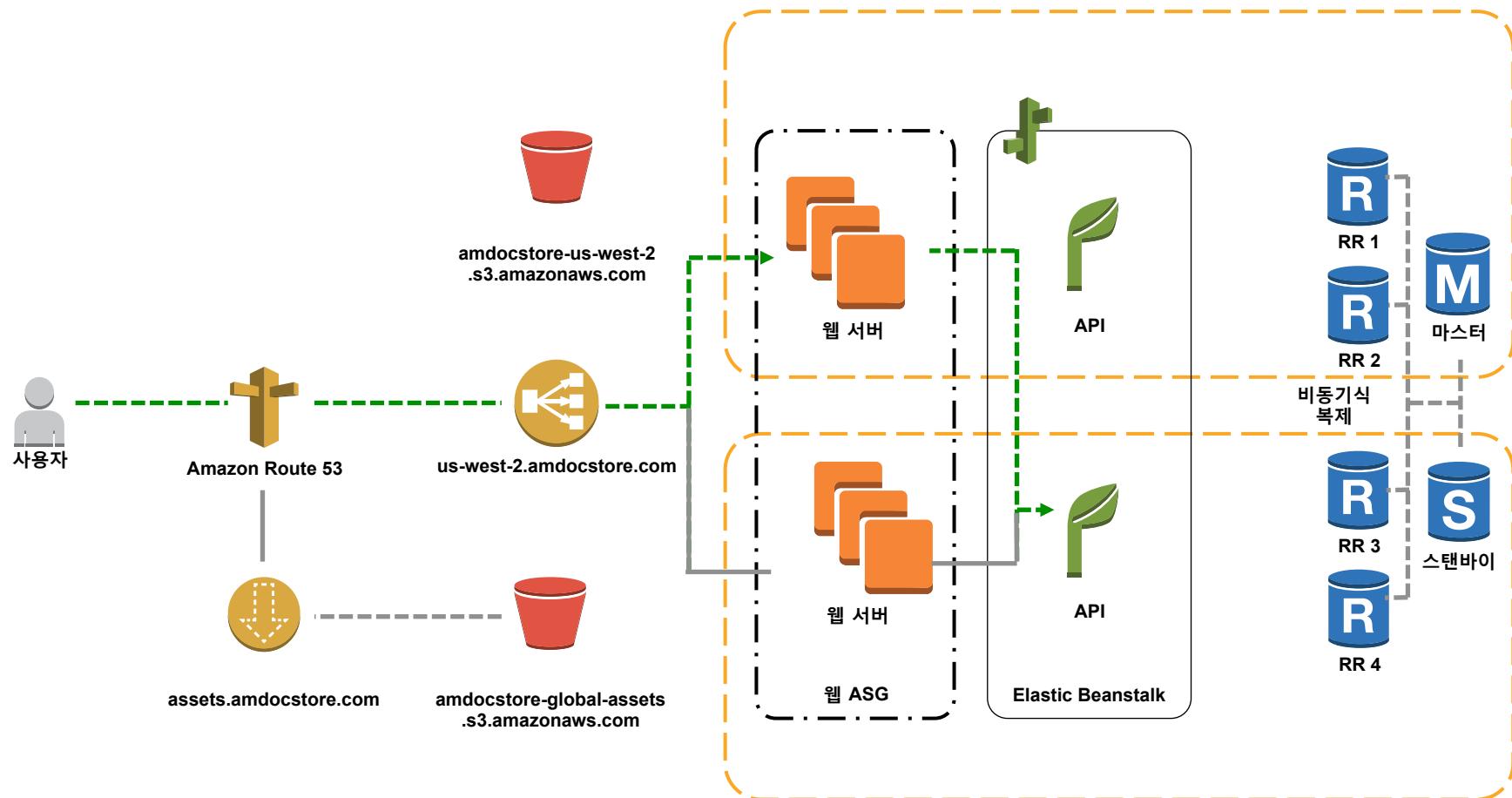
문서 조회 및 다운로드: 인증 받은 사용자가 자신의 문서를 조회하기 위한 웹 사이트 URL을 요청합니다.

DocStore



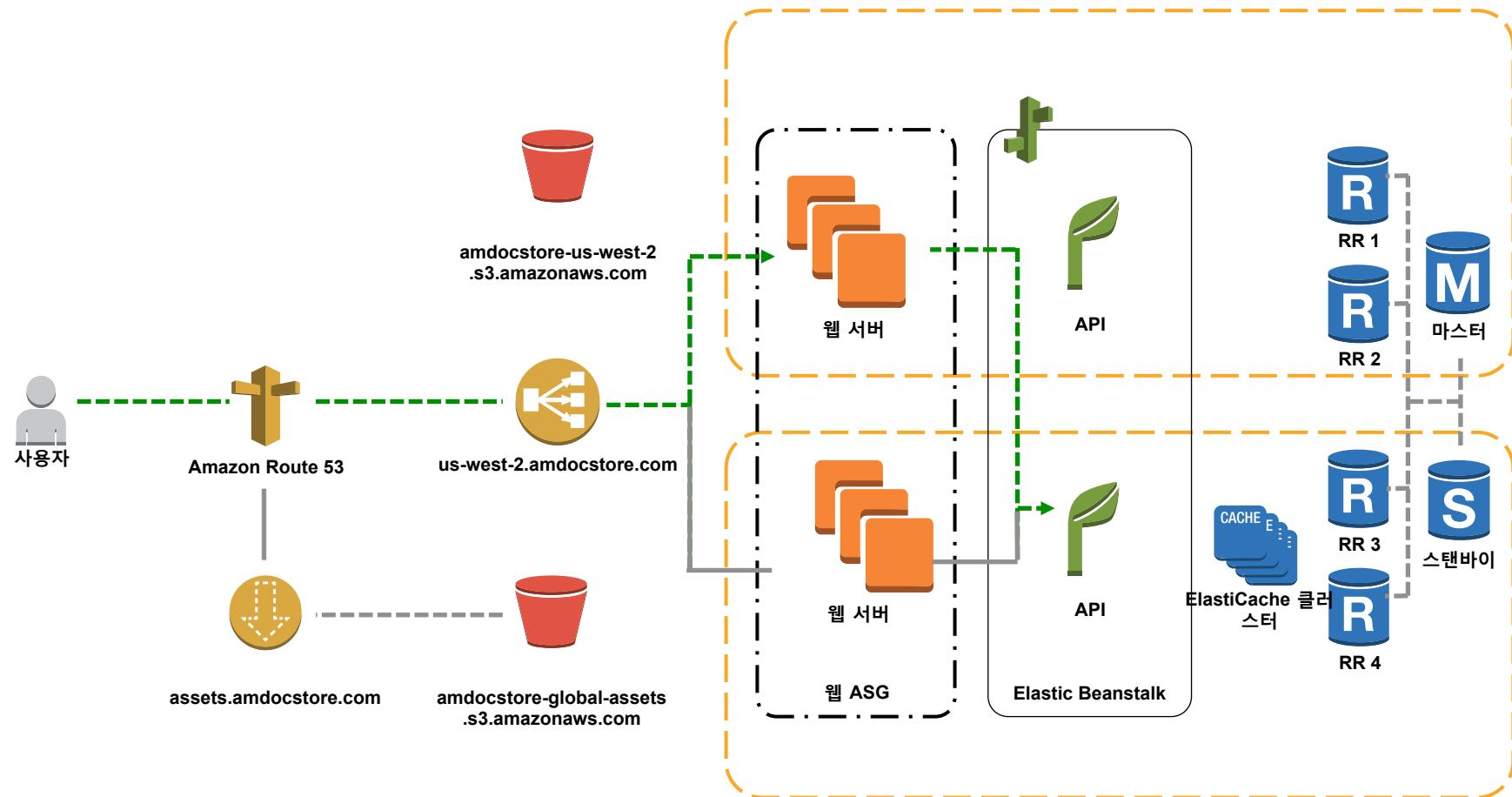
문서 조회 및 다운로드: 관계형 데이터베이스의 읽기 성능을 향상시키기 위해 복수의 RDS 읽기 전용 복제가 프로비저닝되어 있습니다. 읽기 전용 복제는 사용자가 소유한 문서 목록을 선택하는 API와 같은 읽기 작업에 적합합니다.

DocStore



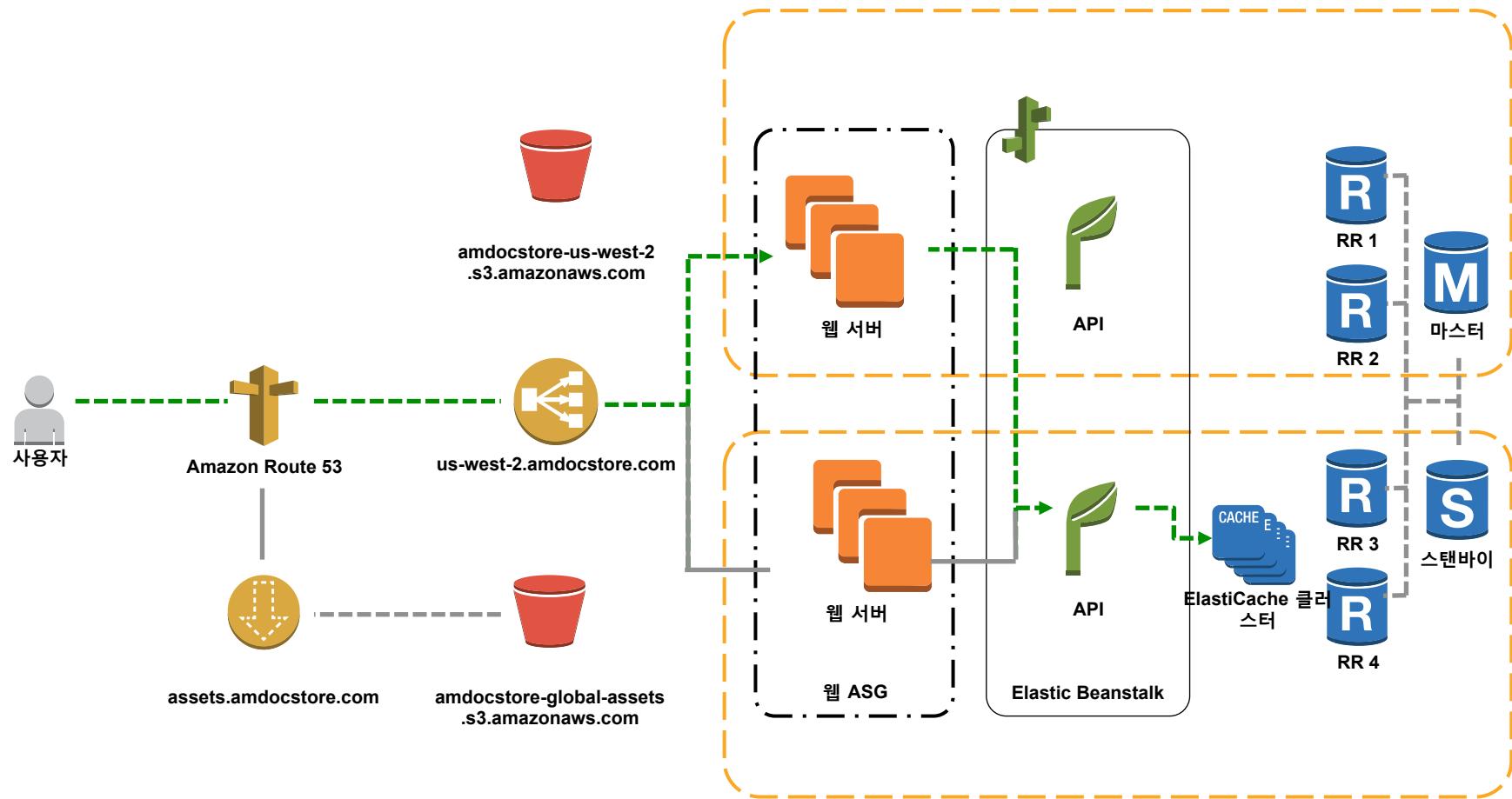
문서 조회 및 다운로드: 읽기 전용 복제 앞단의 ElastiCache
클러스터는 사용자의 문서 목록 조회 시 추가적인 성능 향상
을 제공합니다.

DocStore



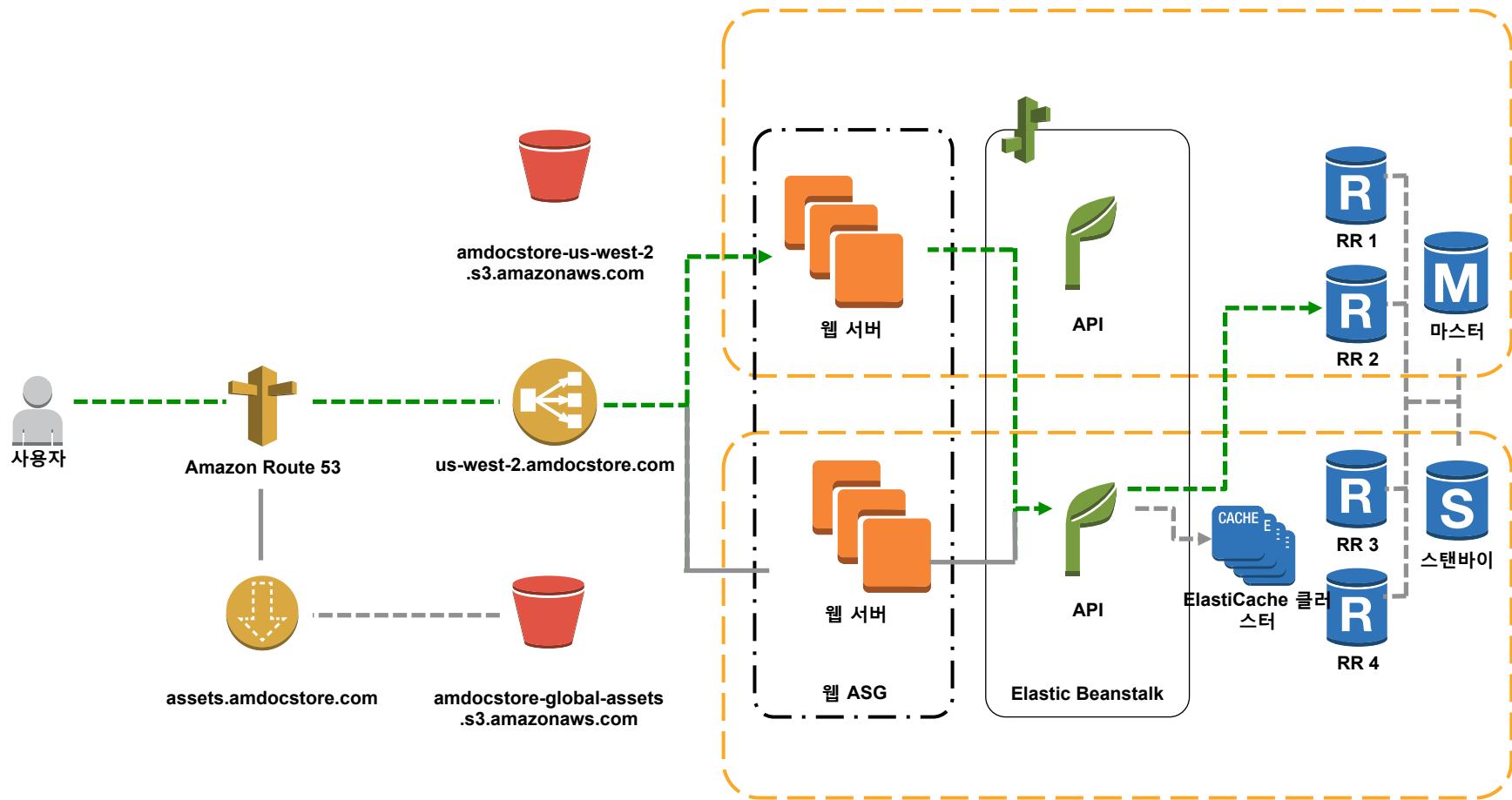
문서 조회 및 다운로드: API는 먼저 캐시에서 사용자 문서의 최근 목록을 확인합니다.

DocStore



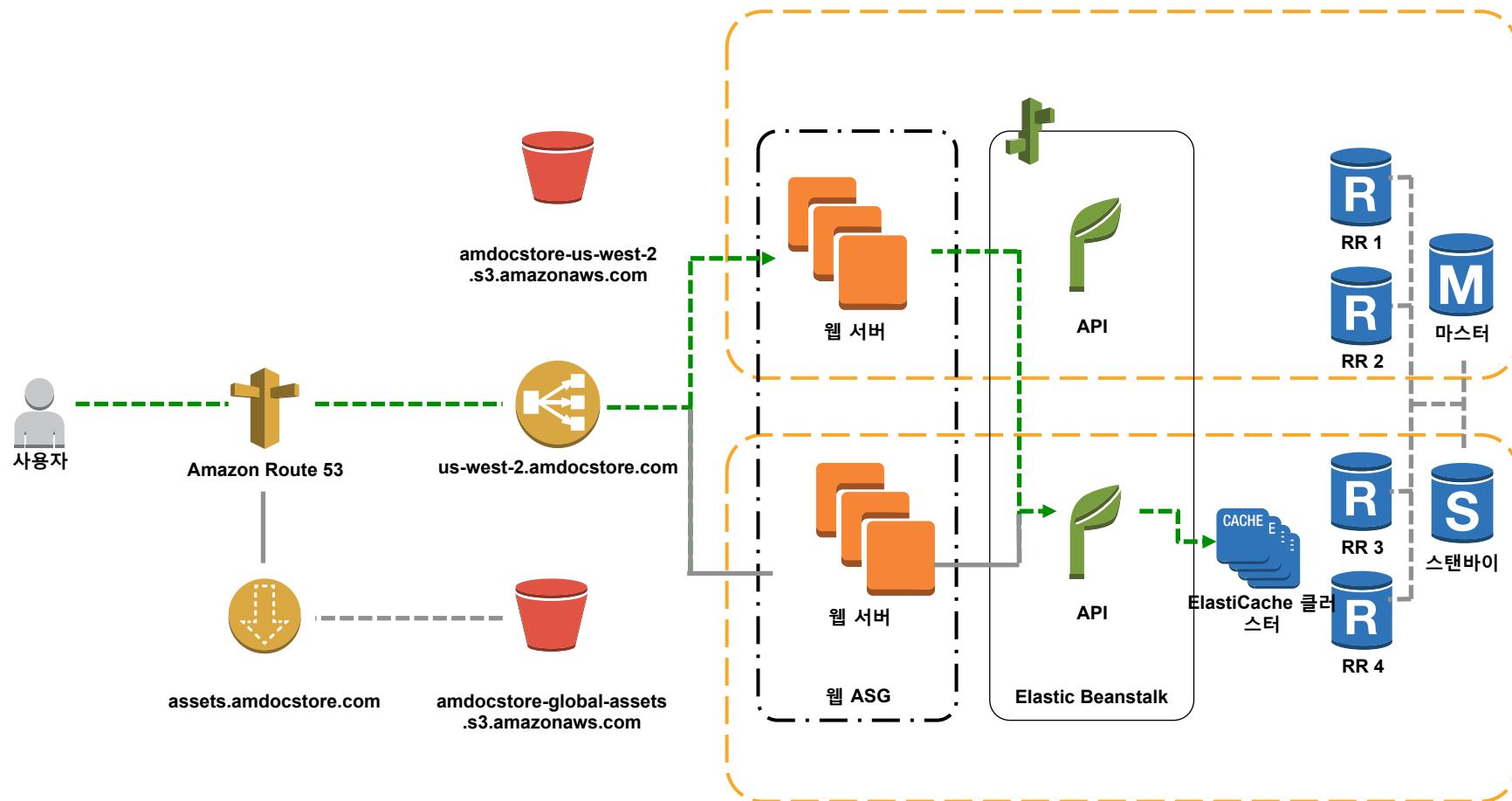
문서 조회 및 다운로드: 캐싱된 데이터가 없으면, API는 읽기 전용 복제 중 하나를 선택해 문서 목록을 조회하기 위한 질의를 합니다.

DocStore



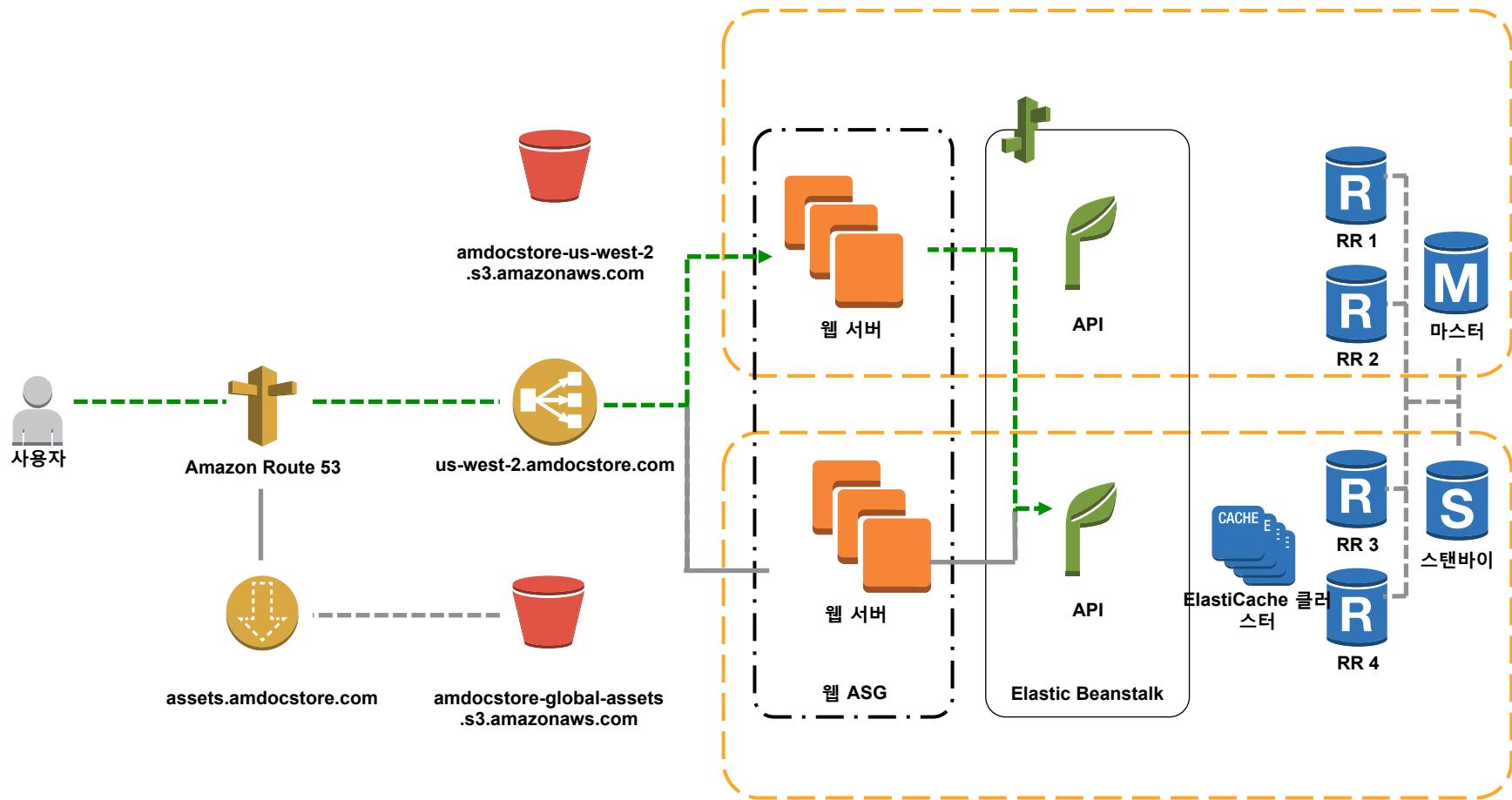
문서 조회 및 다운로드: 질의 결과는 이후 액세스를 위해 캐시에 저장됩니다.

DocStore



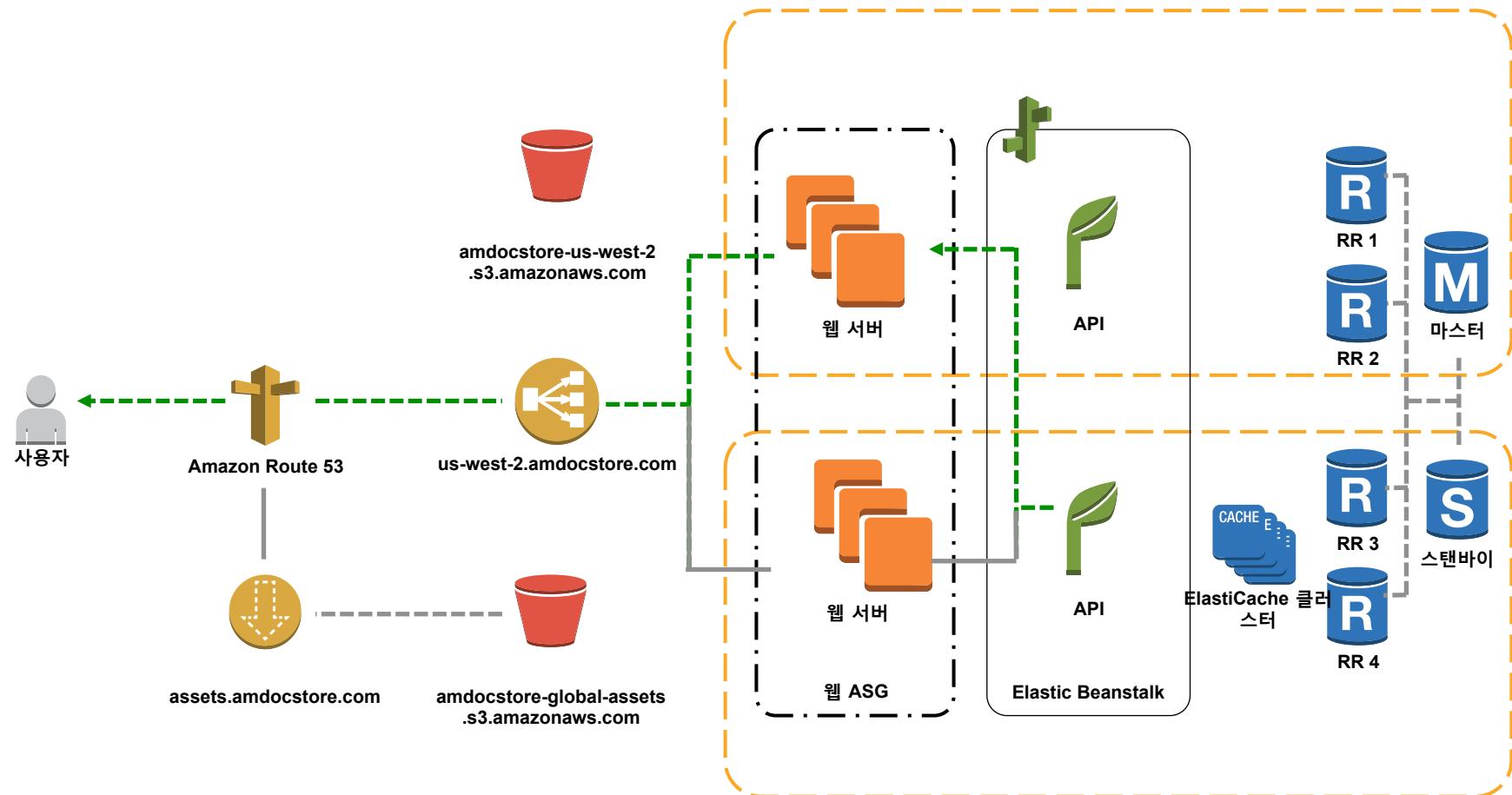
문서 조회 및 다운로드: API는 각 문서에 대해 안전하고 자동으로 만료되는 서명된 URL을 생성하여, 사용자가 S3로부터 직접 문서를 다운로드할 수 있도록 합니다.

DocStore



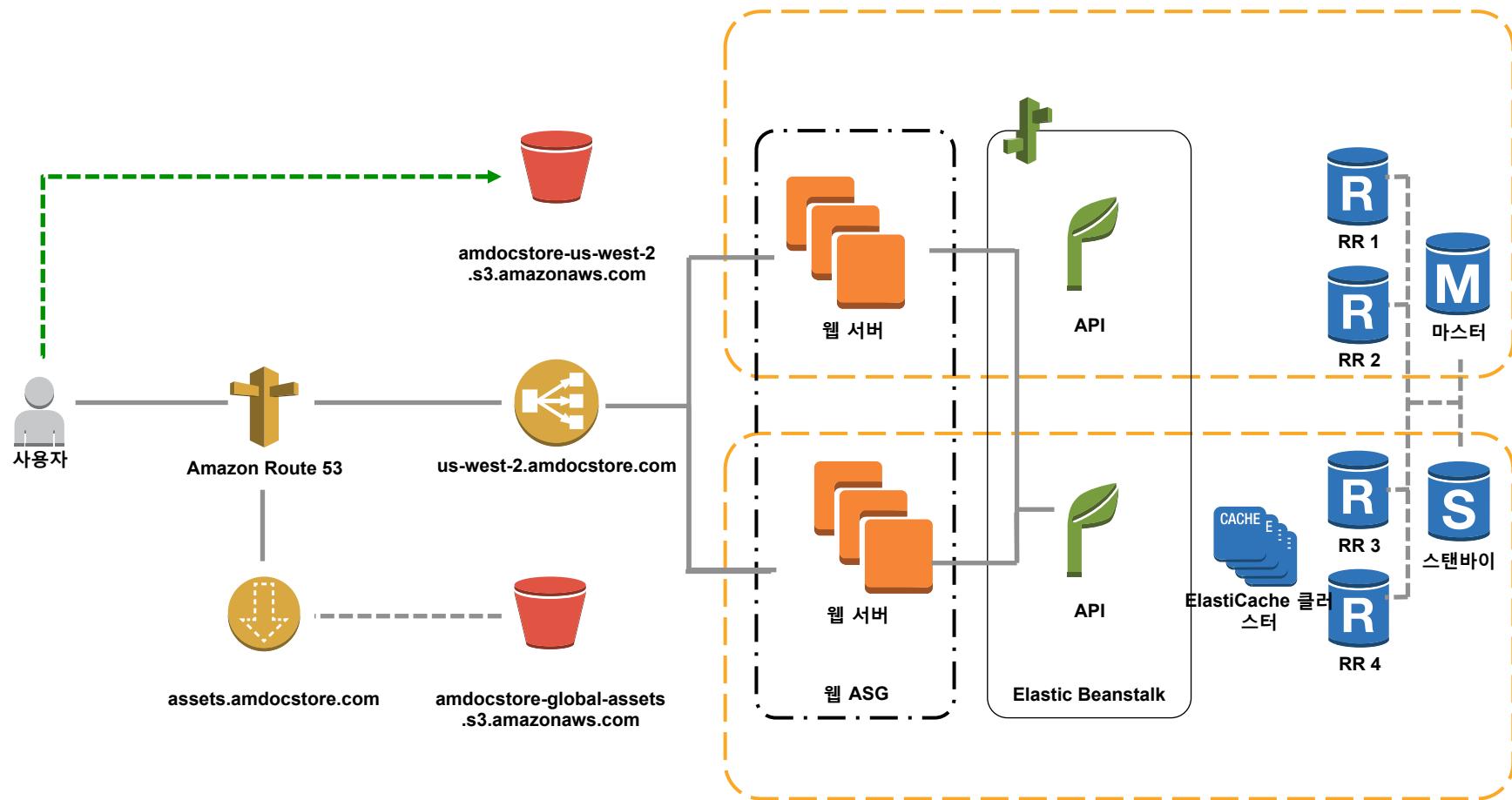
문서 조회 및 다운로드: 문서 목록(안전한 직접 다운로드를 위한 링크 포함)이 사용자의 브라우저로 전달됩니다.

DocStore



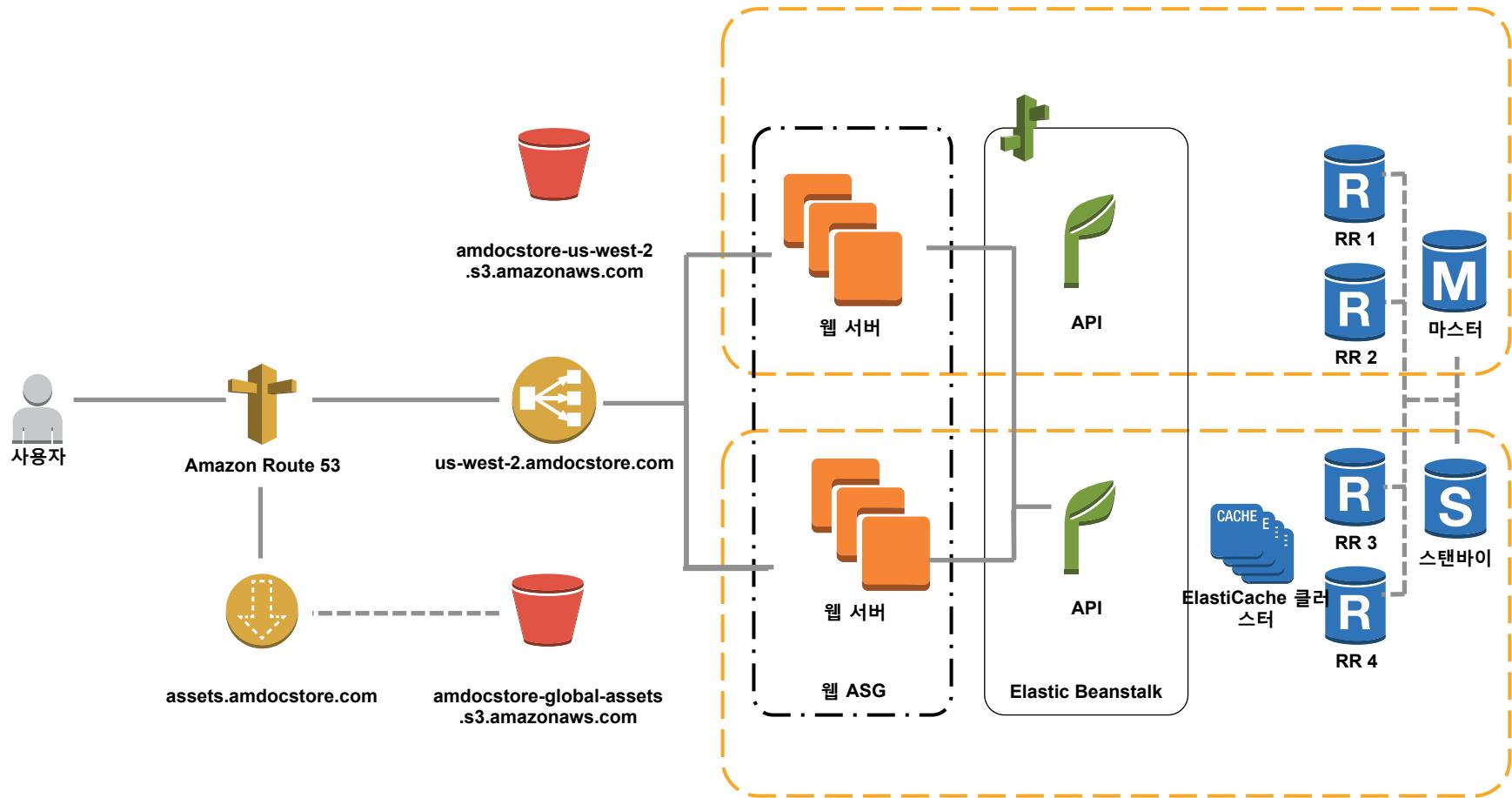
문서 조회 및 다운로드: 사용자는 S3로부터 직접 문서를 다운로드할 수 있습니다.

DocStore



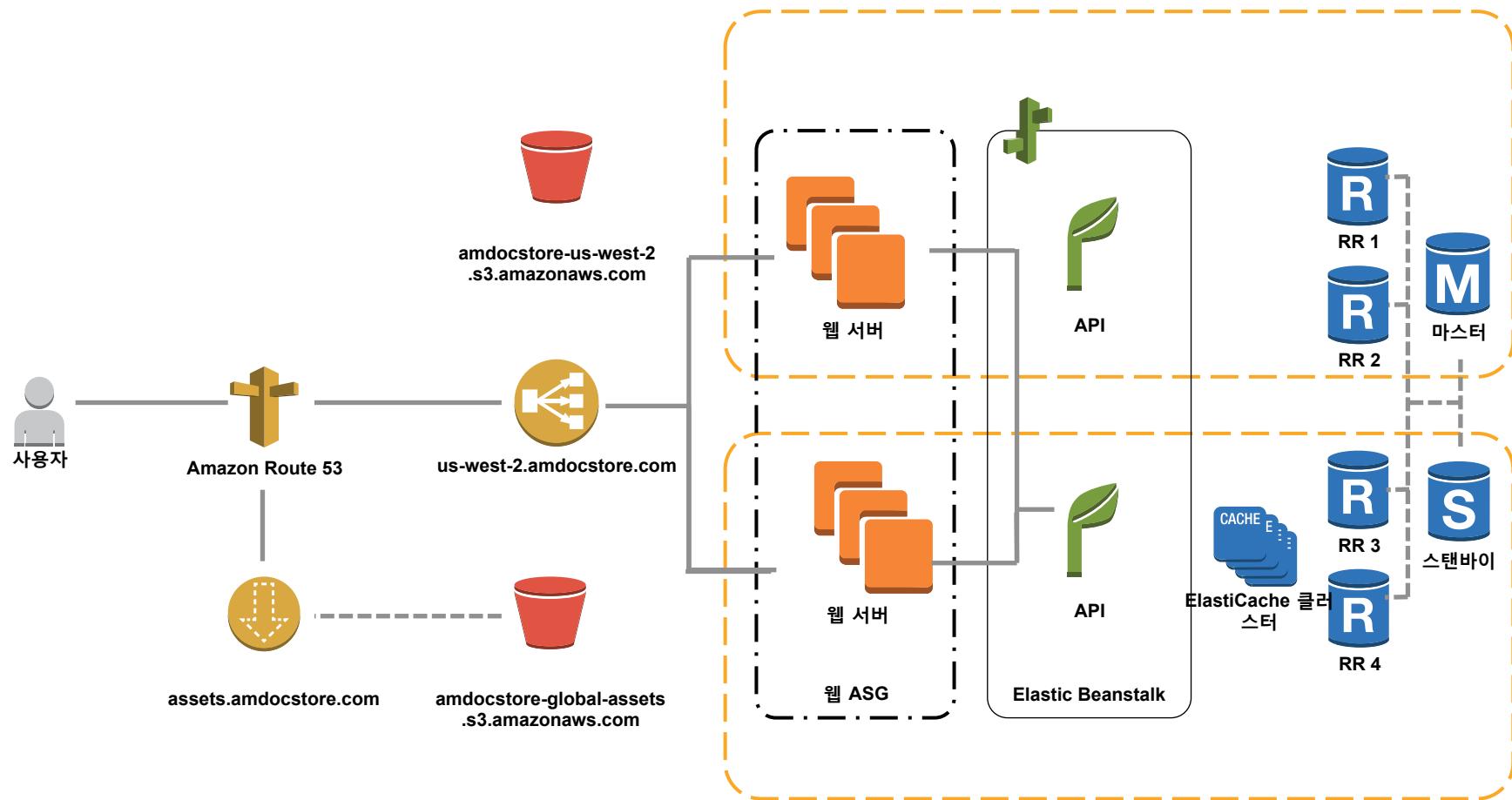
이제 보안을 위한 옵션을 살펴보겠습니다.

DocStore



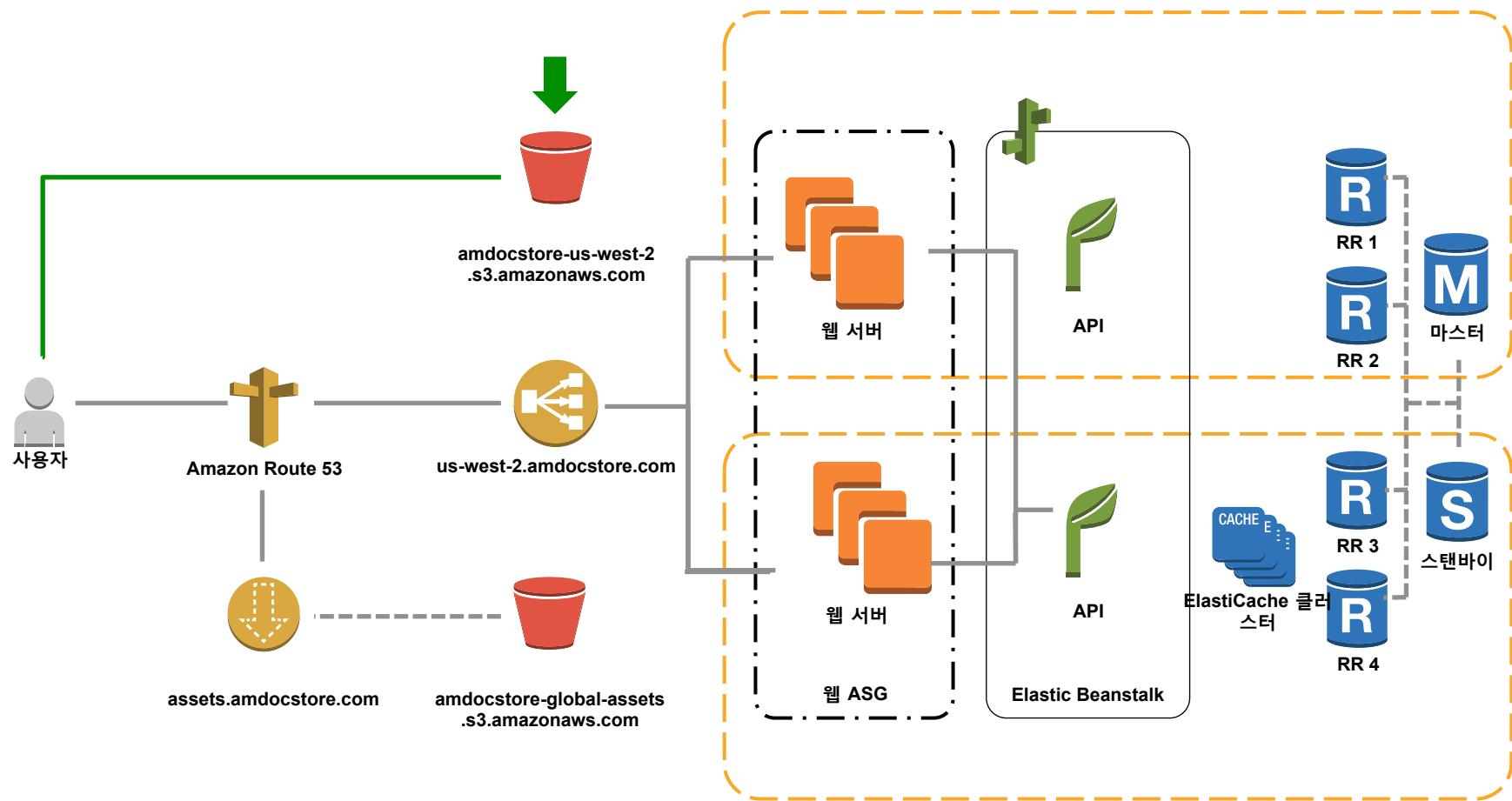
보안: 본 아키텍처 내 어디에서 보안이 구현될 수 있을까요?

DocStore



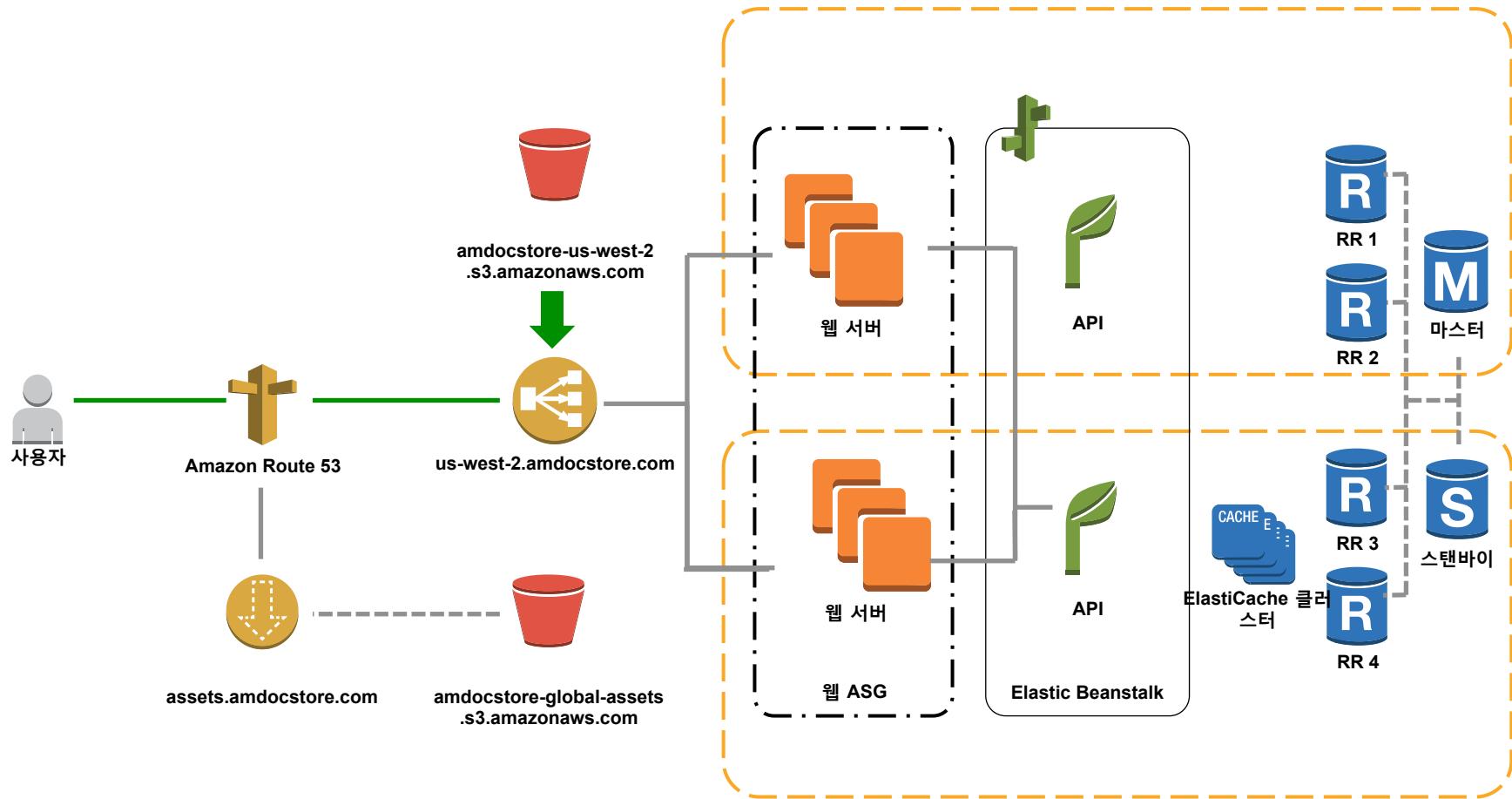
보안: 사용자는 S3로부터 직접 문서를 다운로드할 때 SSL을 통하는 서명되고 안전한 URL을 사용합니다.

DocStore



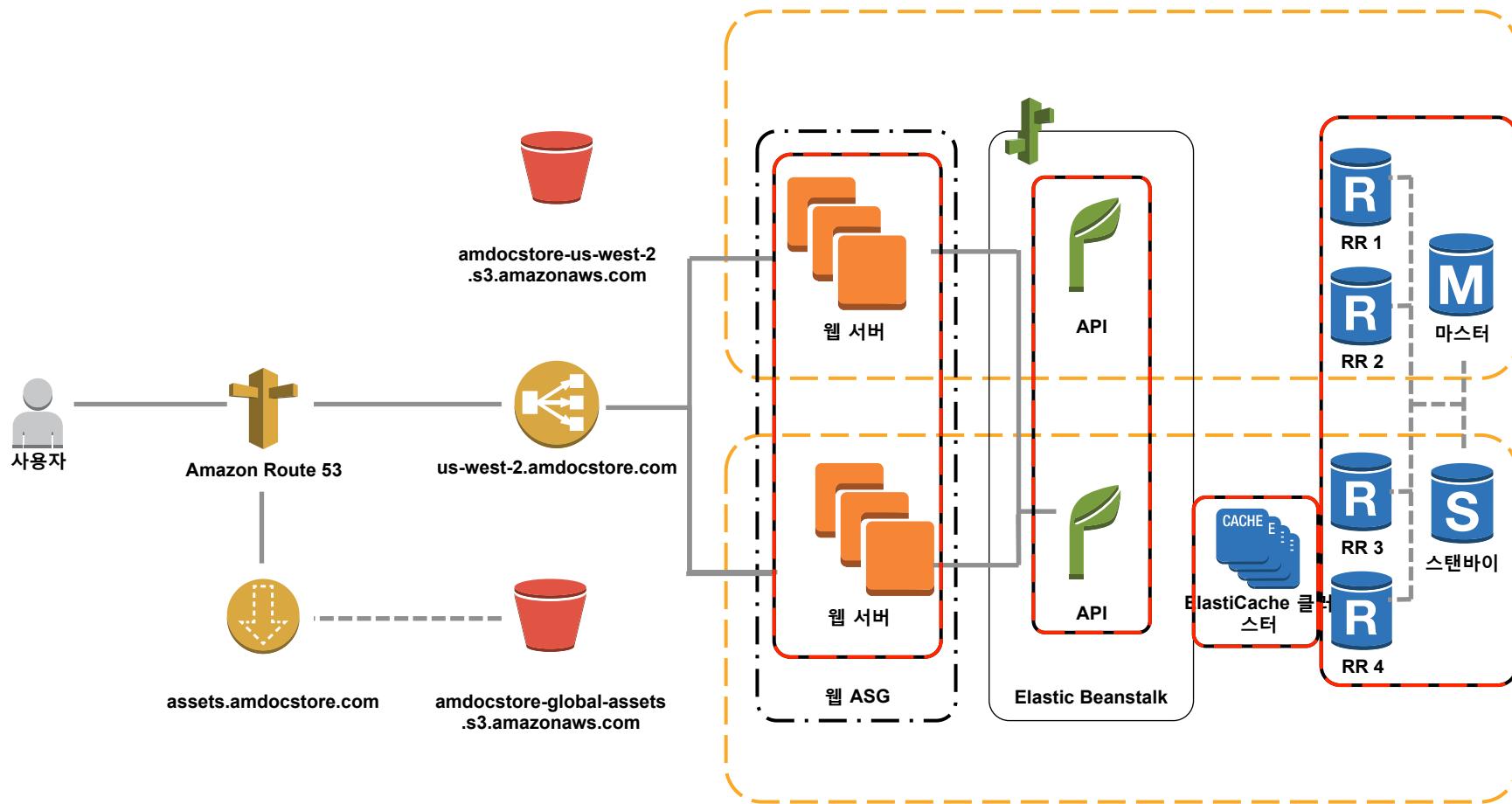
보안: 사용자와 ELB 간의 연결에도 SSL이 적용됩니다.

DocStore



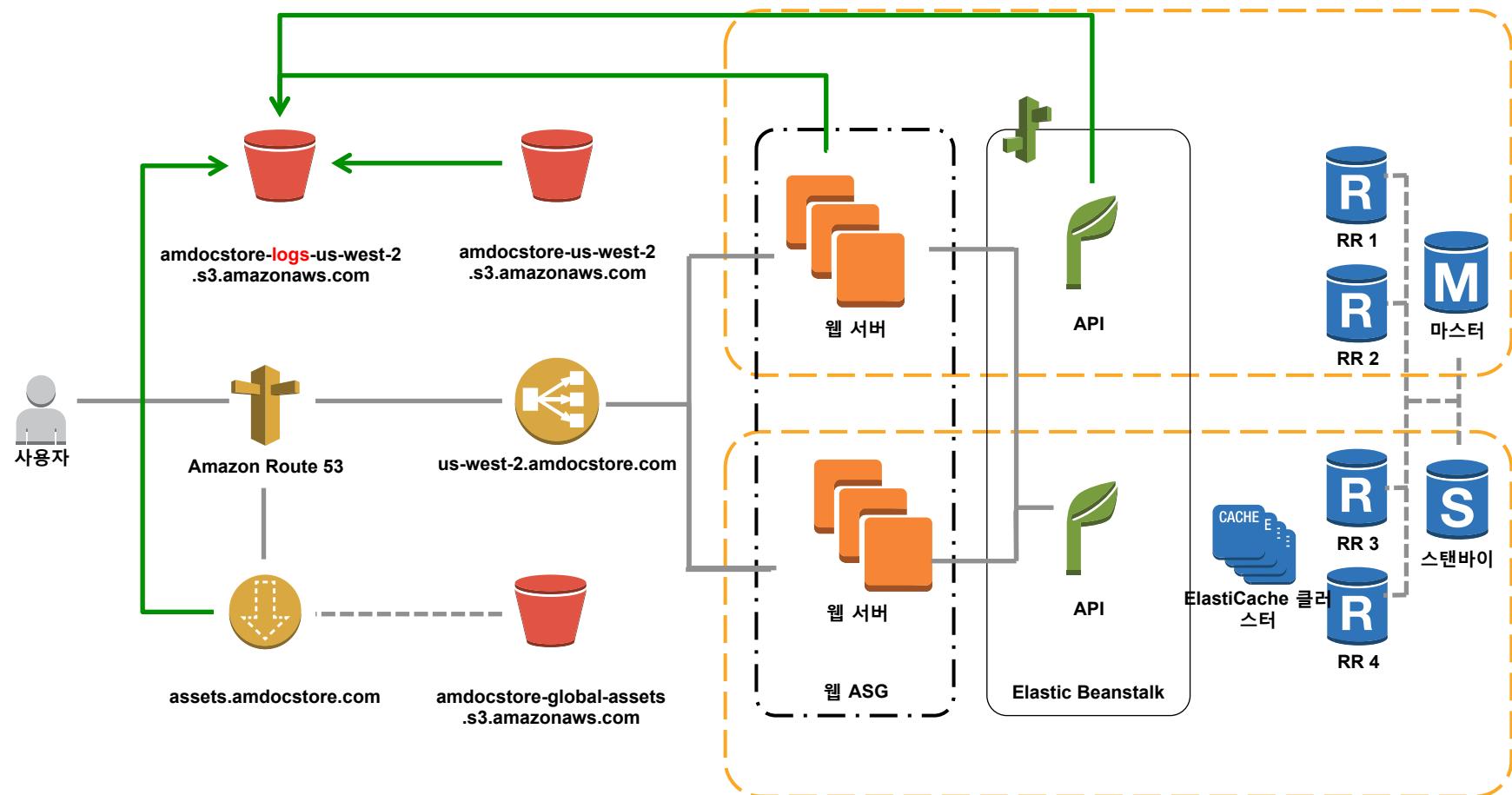
보안: 보안 그룹은 인바운드 트래픽을 제한합니다.

DocStore



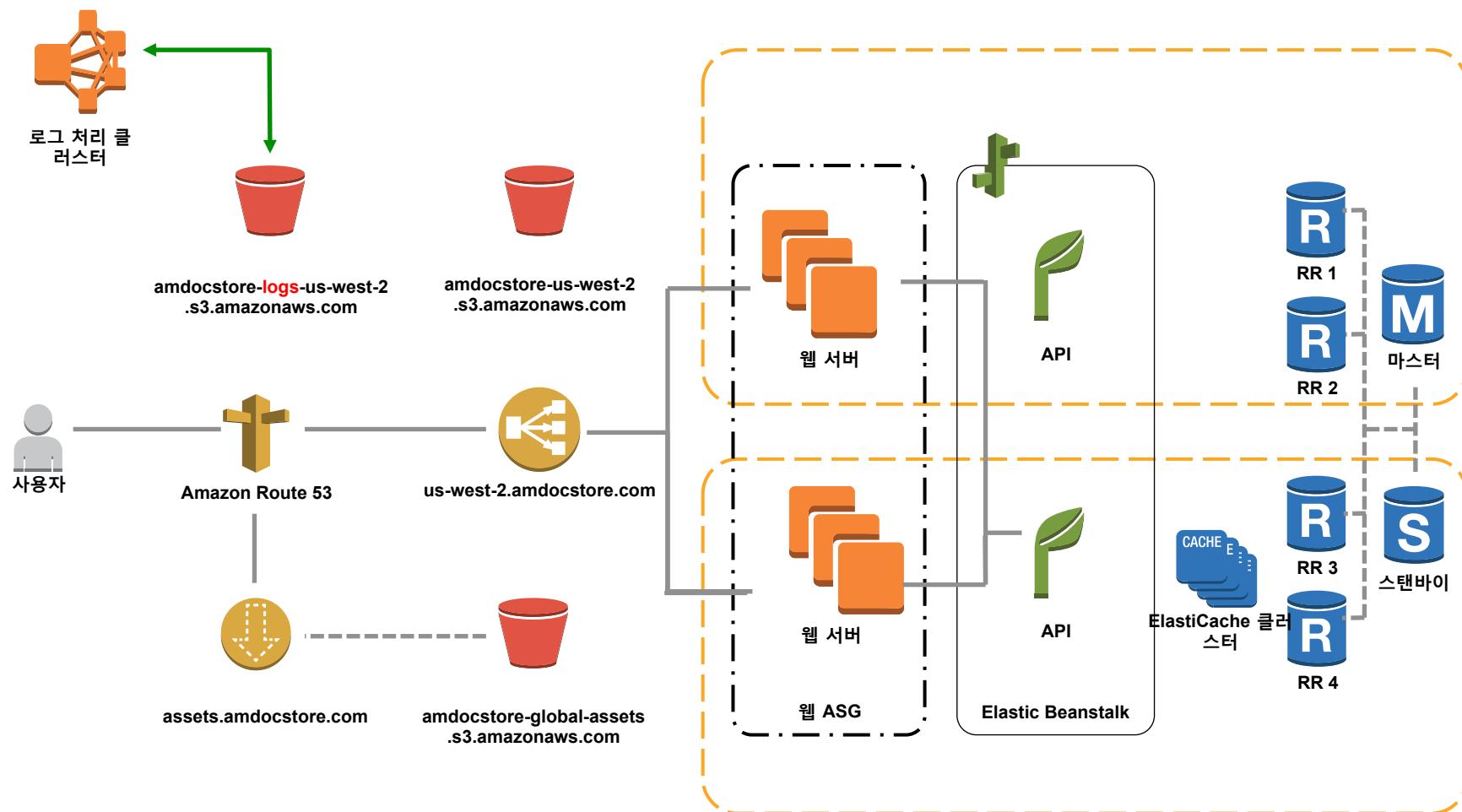
보안: CloudFront, EC2, S3에 대한 액세스 로그는 S3에 저장 됩니다.

DocStore



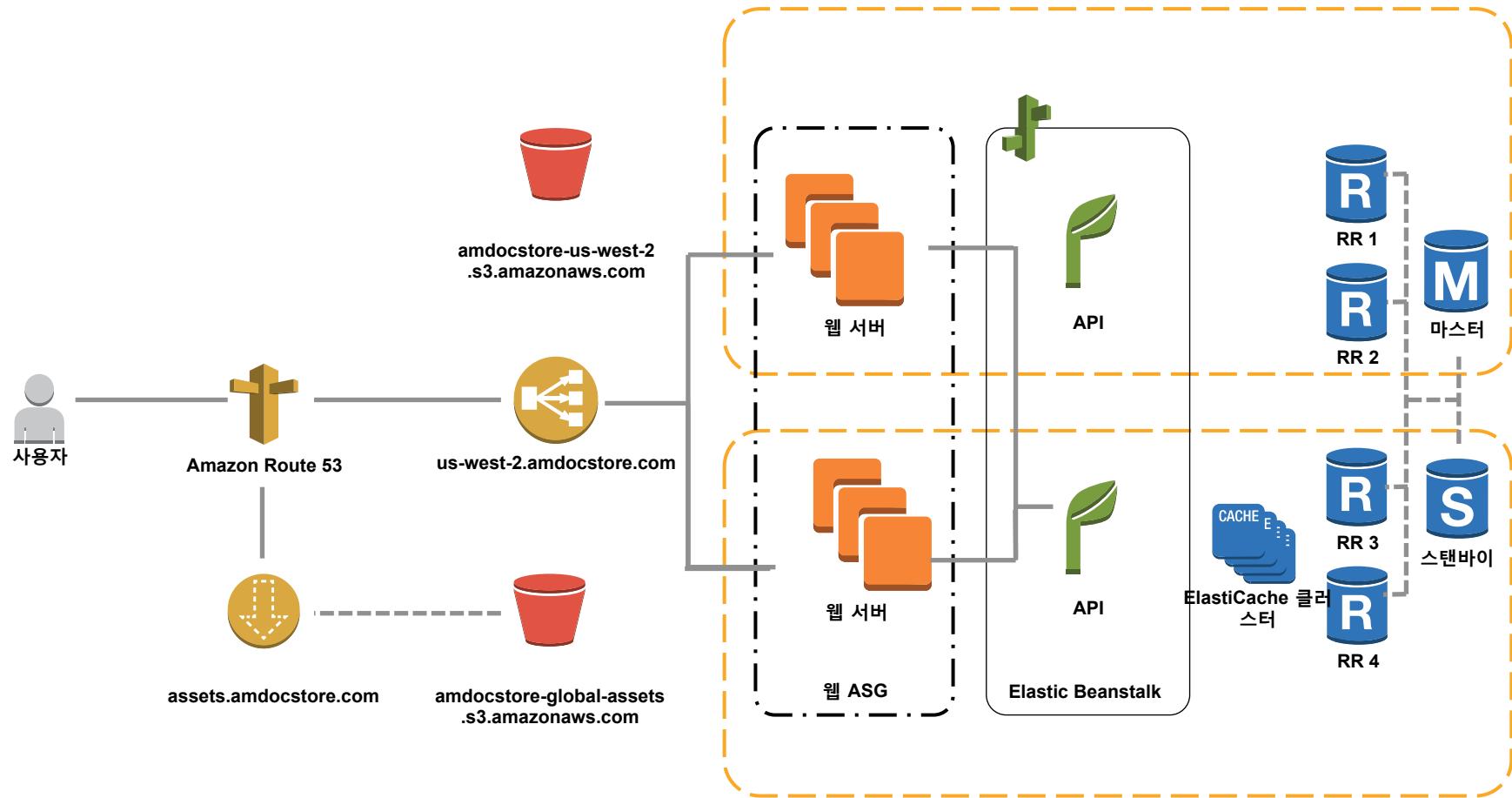
보안: 로그는 주기적으로 Elastic MapReduce 클러스터를 사용해 처리됩니다.

DocStore



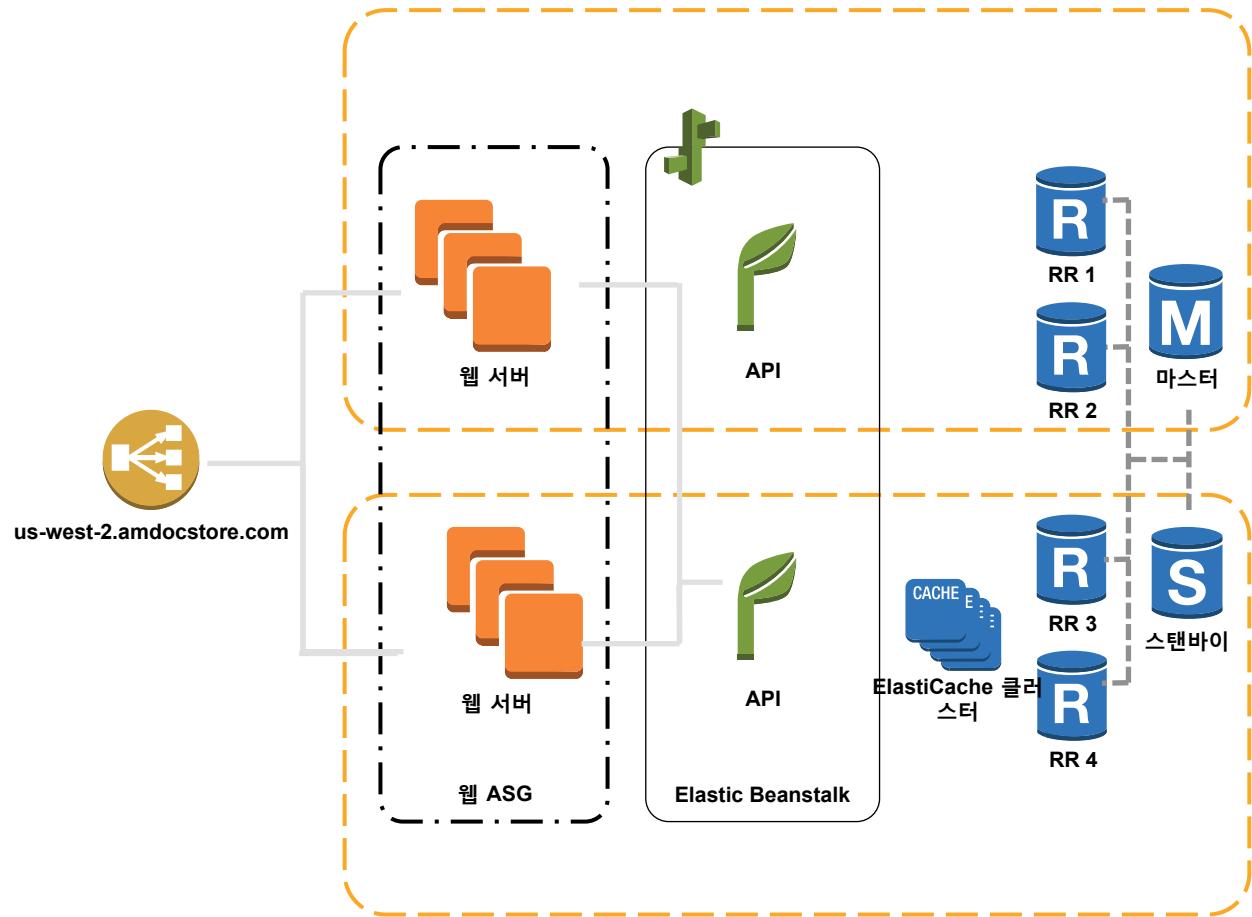
이제 네트워크 환경에 대해 살펴보겠습니다.

DocStore



이제 네트워크 환경에 대해 살펴보겠습니다.

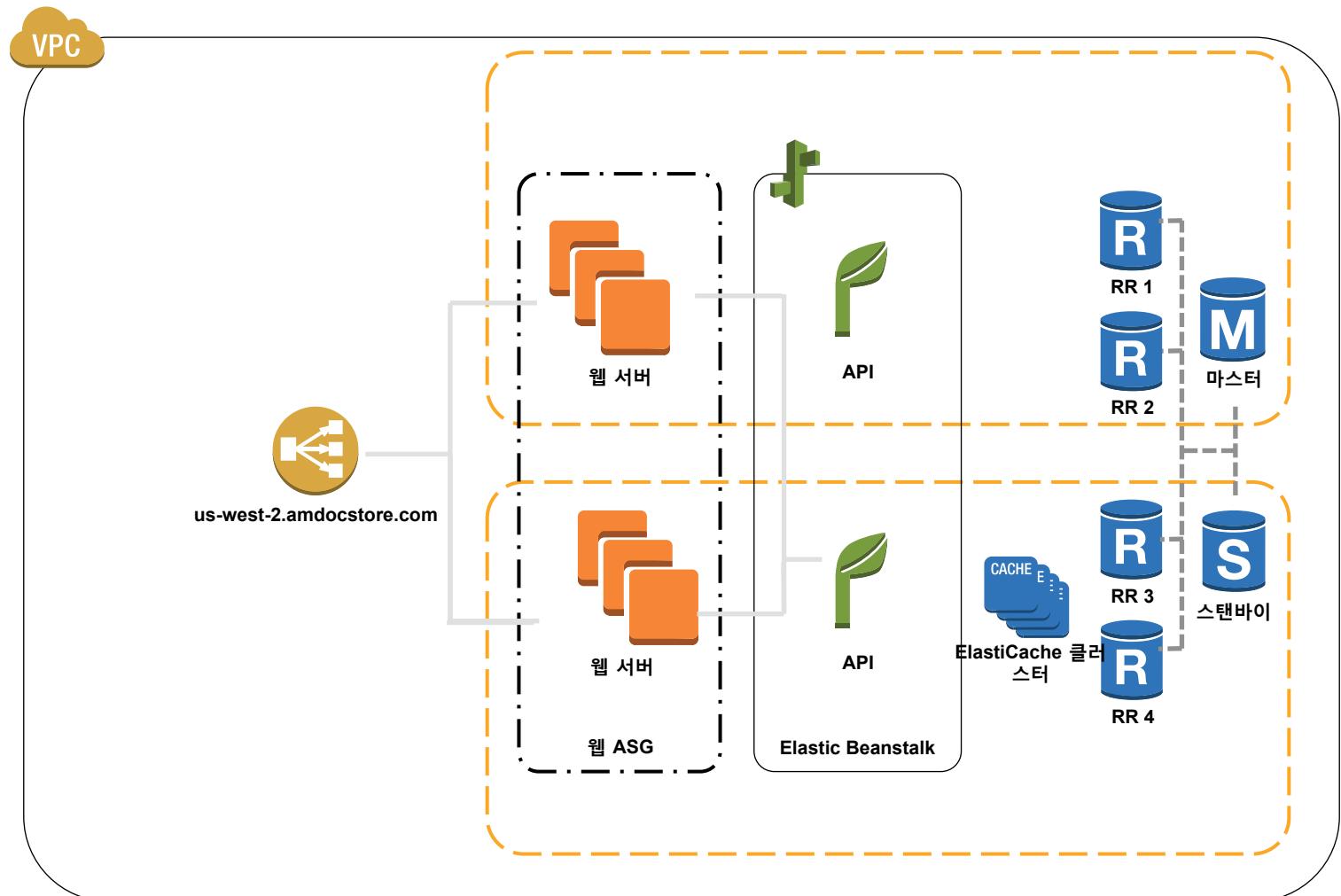
DocStore



VPC: 본 VPC는 us-west-2 내에서 10.0.0.0/16 CIDR 블록을
가집니다.

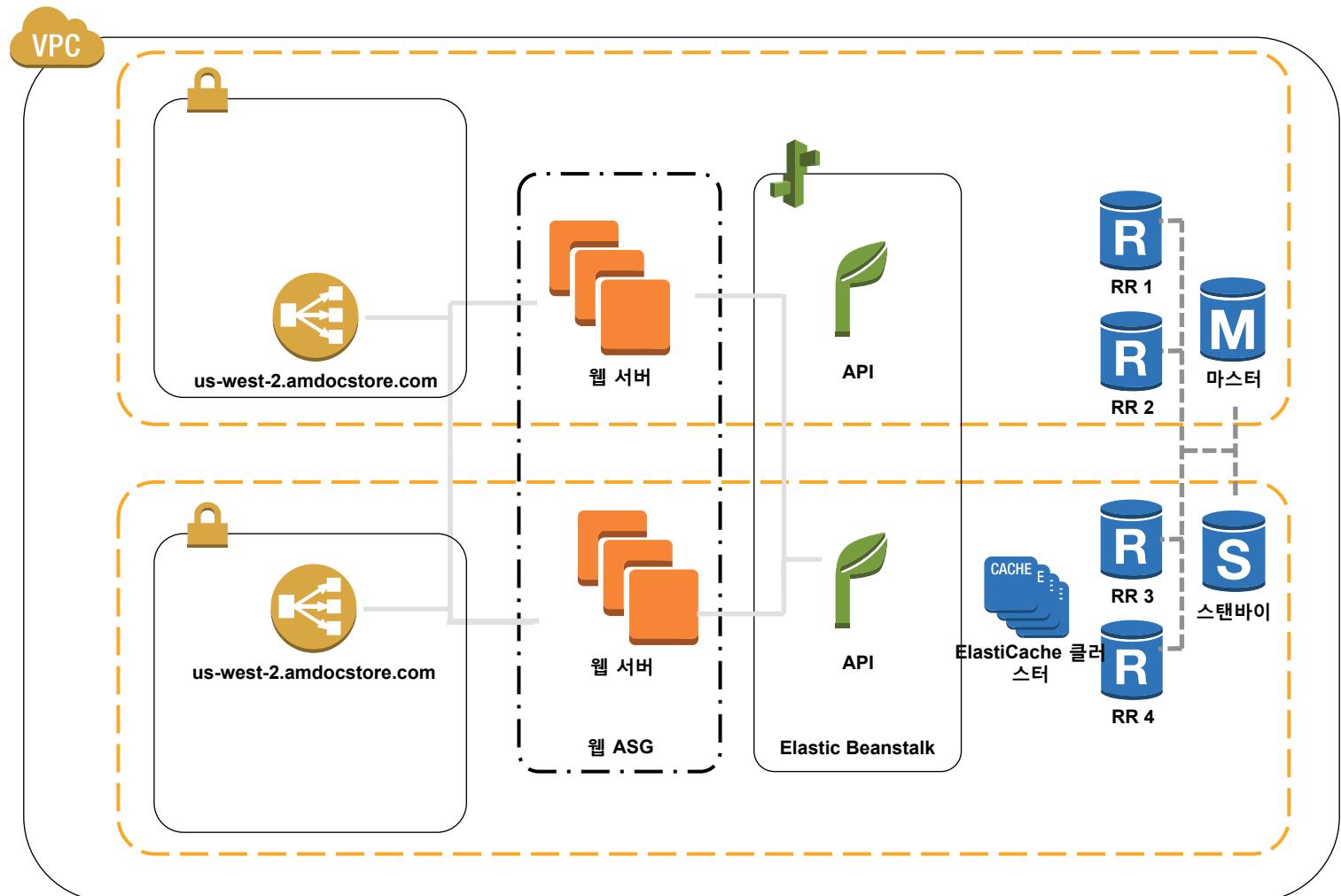


DocStore



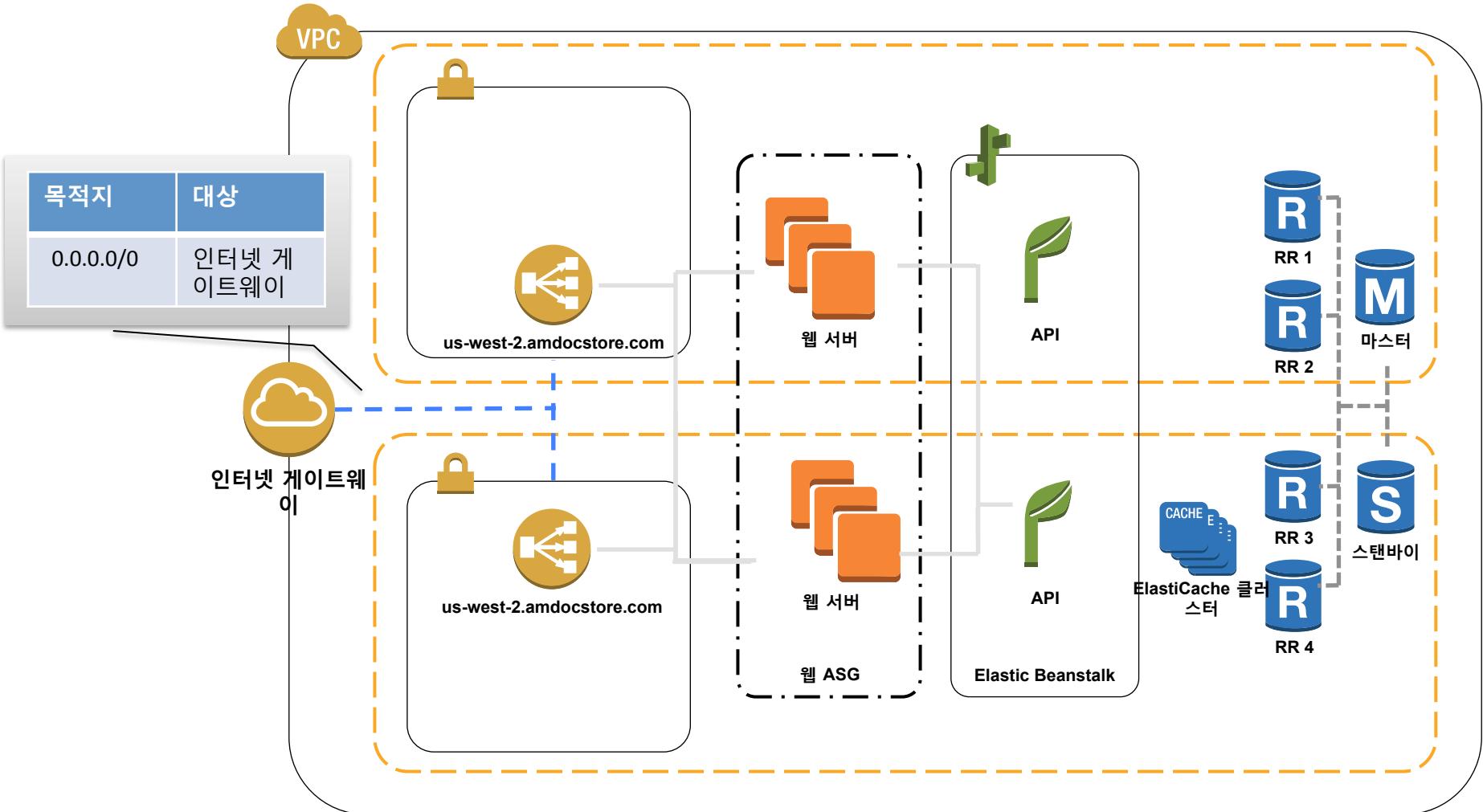
VPC: ELB가 구동되는 두 개의 서브넷을 정의합니다(주의 :
ELB가 서브넷별로 표시되어 있지만 여전히 하나의 ELB입니다).

DocStore



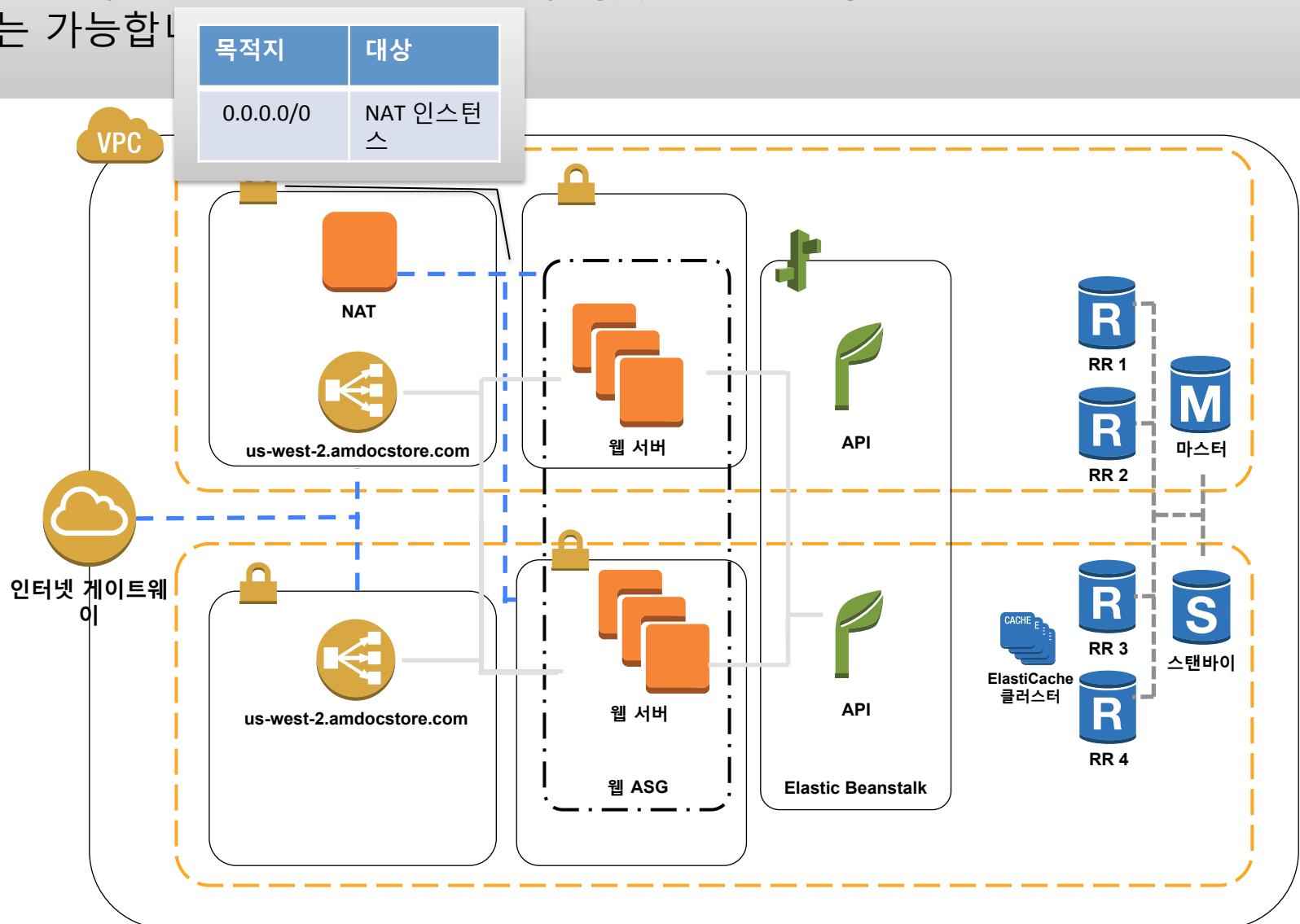
VPC: ELB 서브넷은 퍼블릭이어야 하므로 **인터넷 게이트웨이**를 장착하고 **라우팅 테이블 항목**을 정의합니다.

DocStore



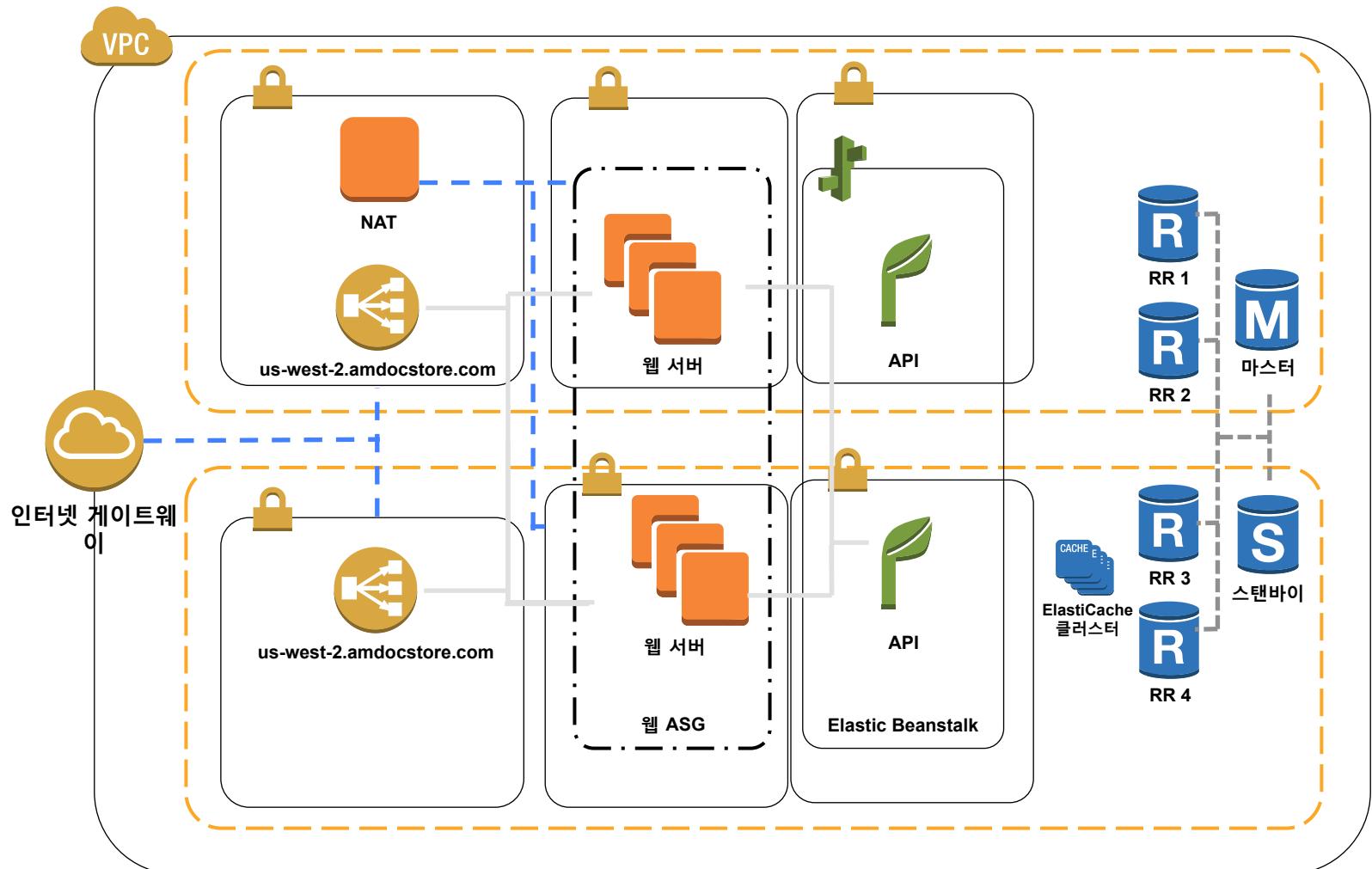
VPC: 웹 서버는 프라이빗 서브넷 내에서 구동하지만 퍼블릭 서브넷 내 NAT 구동 EC2 인스턴스를 통해 **아웃바운드 인터넷 액세스**는 가능합니다.

DocStore



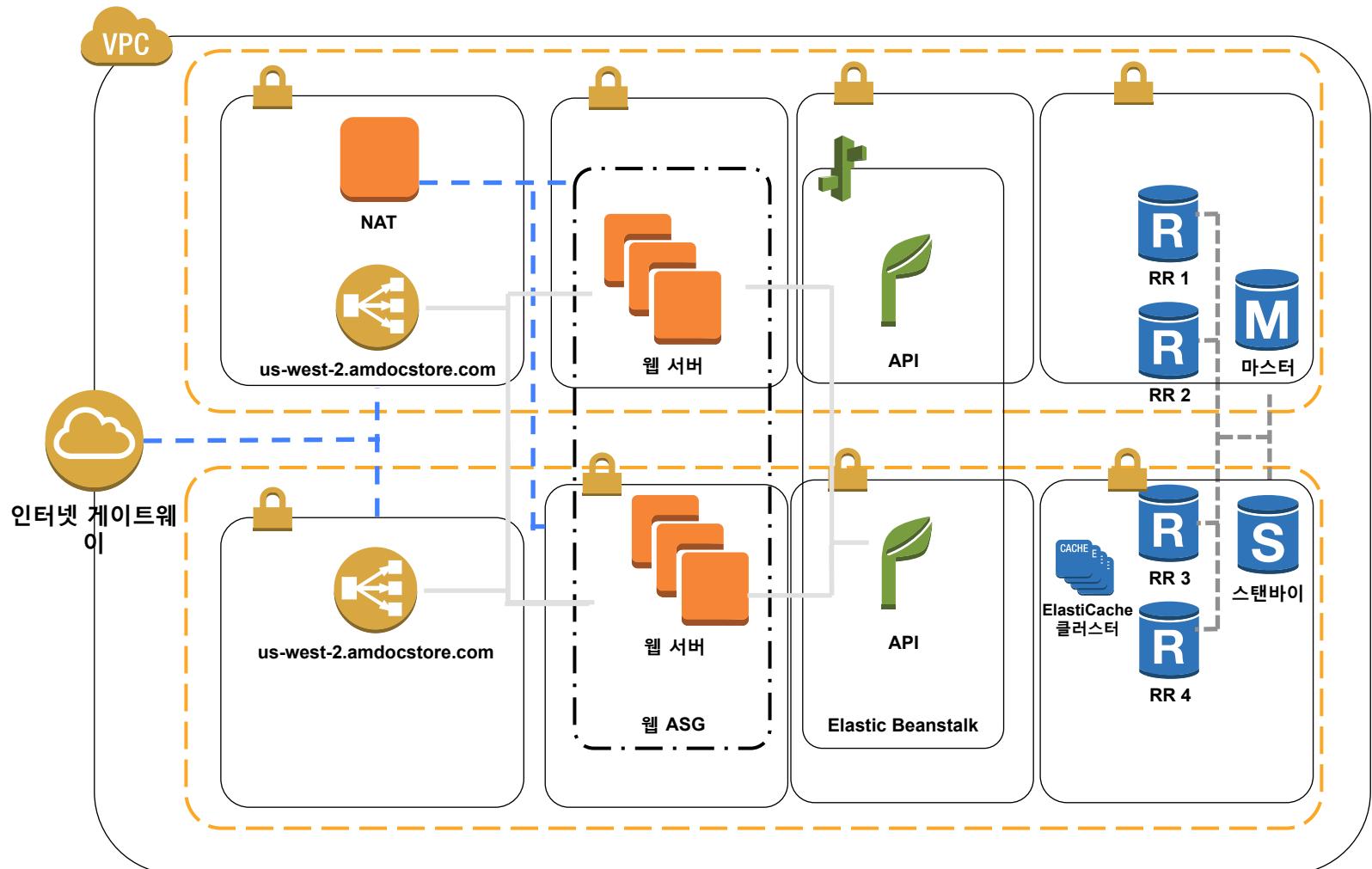
VPC: Elastic Beanstalk에서 구동하는 DocStore API는
ELB를 위한 퍼블릭 서브넷과 EC2를 위한 프라이빗
서브넷에 배치됩니다.

DocStore



VPC: 마지막으로, RDS와 ElastiCache 인스턴스는 프라이빗 서브넷에 배치됩니다.

DocStore



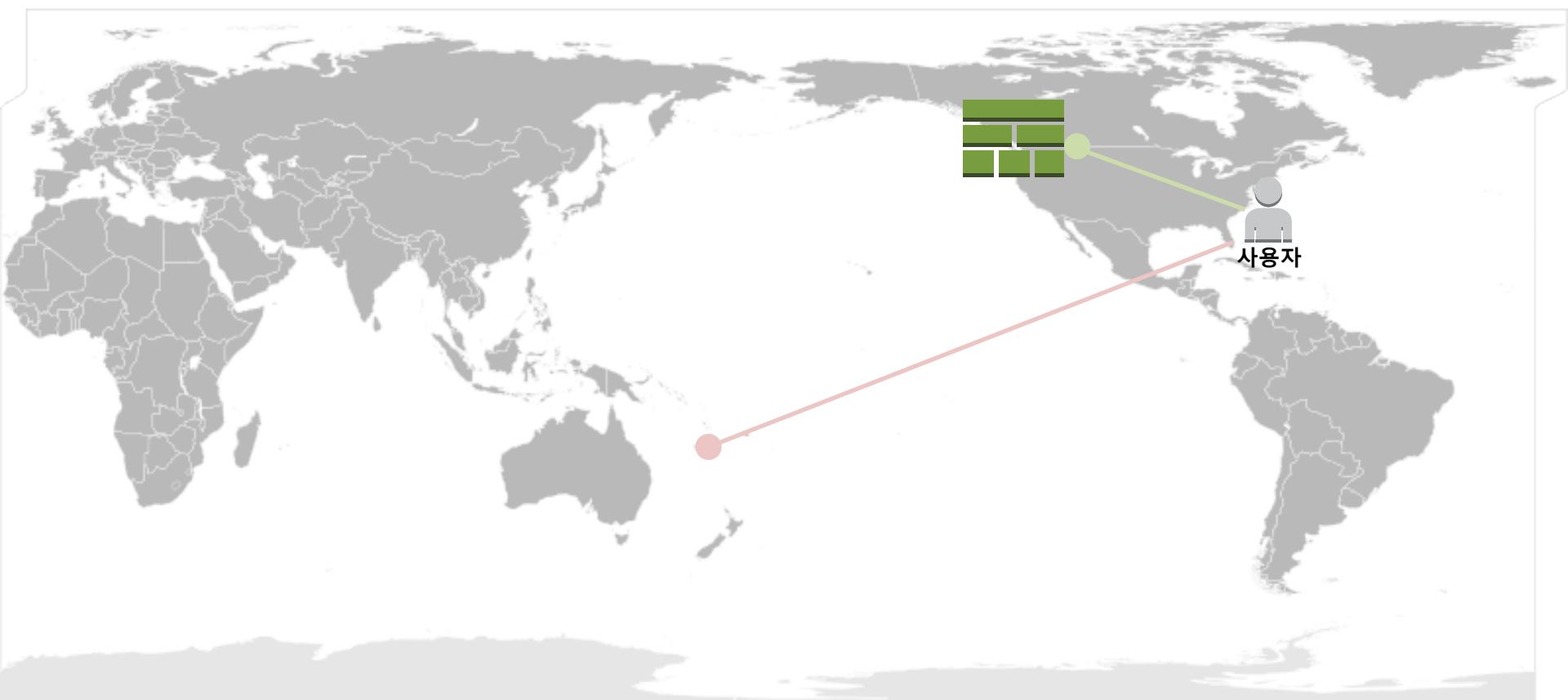
CloudFormation: 전체 애플리케이션은 템플릿 파일로부터 스택으로 배치됩니다.

DocStore



CloudFormation: 샘플 템플릿 파일이 us-west-2에 스택을 배치하는 데 사용되었습니다.

DocStore



CloudFormation: 그리고 ap-southeast-2에도 사용되었습니다.

DocStore



DocStore

#2: 백엔드에서의 배치 프로세싱

- 사용자 계정 동기화
- 스토리지 최대 용량 제한
- 문서 유형을 기반으로 하는 내용 검색을 위한 색인 작업(즉, PDF는 DOCX와 다른 필터 사용 등)
- 시스템 전체 총합을 포함한 상세 대시보드:
 - 문서 개수, 평균 문서 크기, 스토리지 총량

DocStore

#2: 백엔드에서의 배치 프로세싱

- 웹 인터페이스에 의해 발생한 이벤트에 대응



신규 계정
생성 이벤트



신규 계정 생
성



문서 업로드
시작 이벤트



문서 업로드
시작



문서 업로드
완료 이벤트



문서 업로드
완료

DocStore

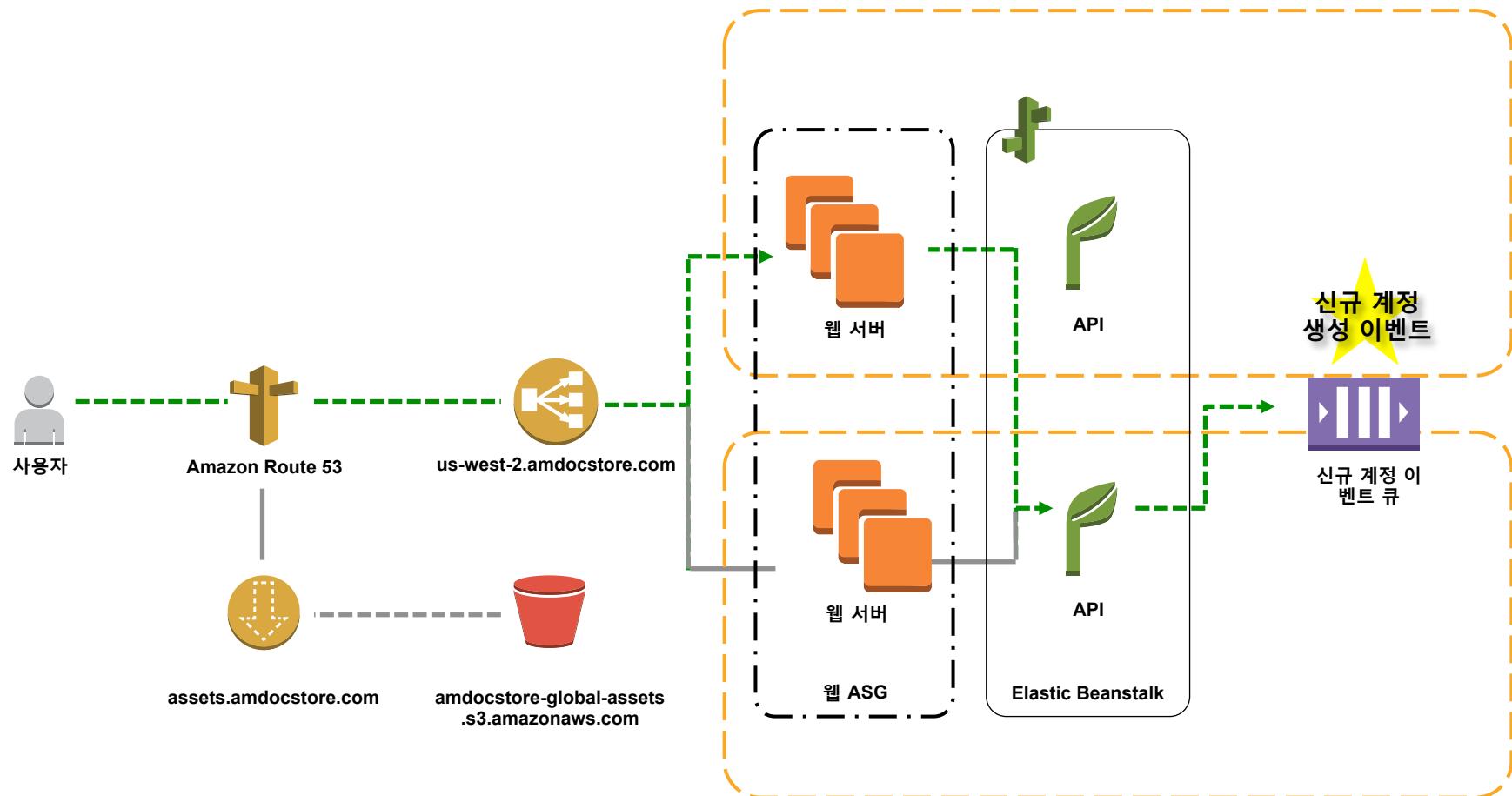
#2: 백엔드에서의 배치 프로세싱

- 웹과 모바일 인터페이스에 의해 발생한 이벤트에 대응



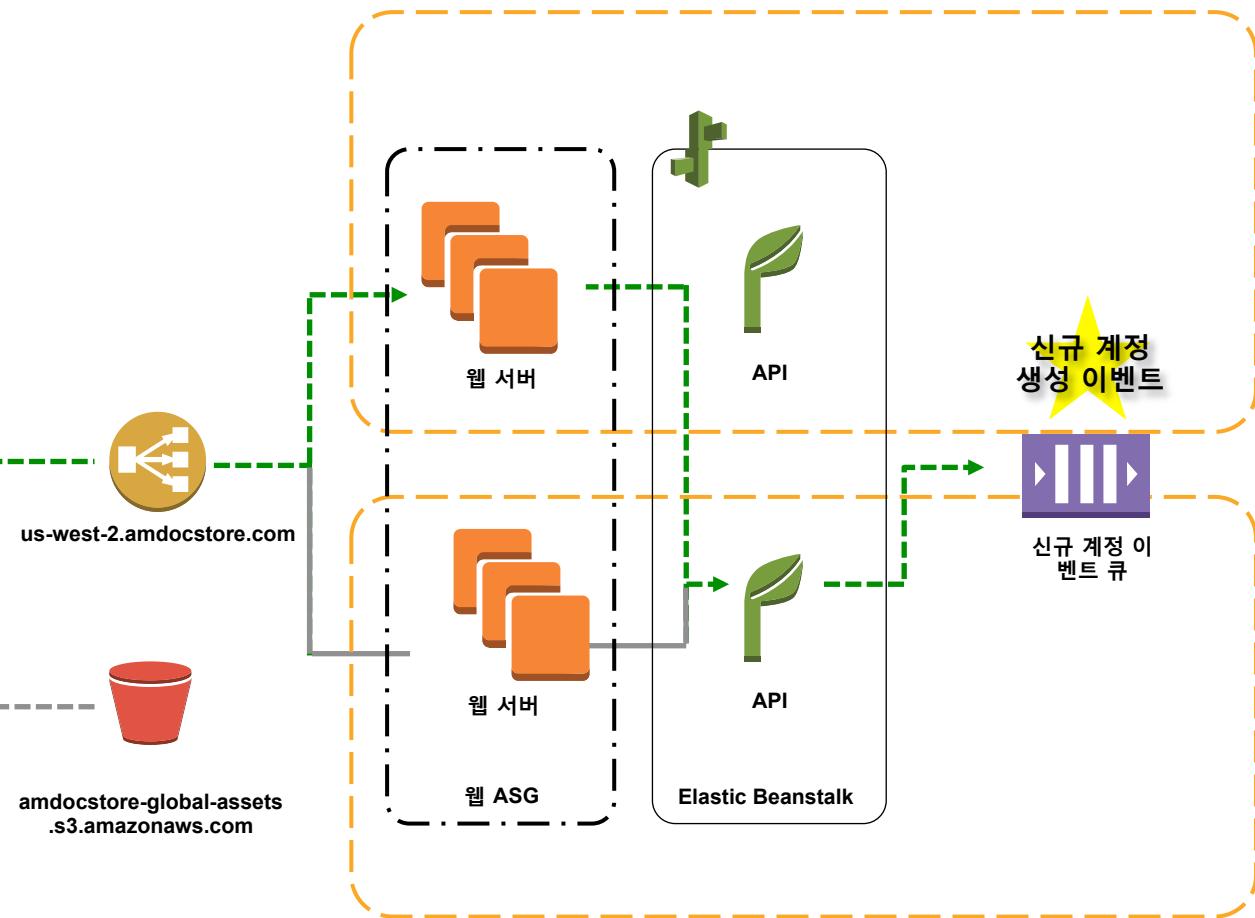
신규 계정 생성 이벤트: 사용자가 새로운 계정을 생성할 때 발생하는 이벤트입니다. 큐에 저장되는 이벤트 메시지는 사용자 이름, 암호, 홈 지역을 포함합니다.

DocStore



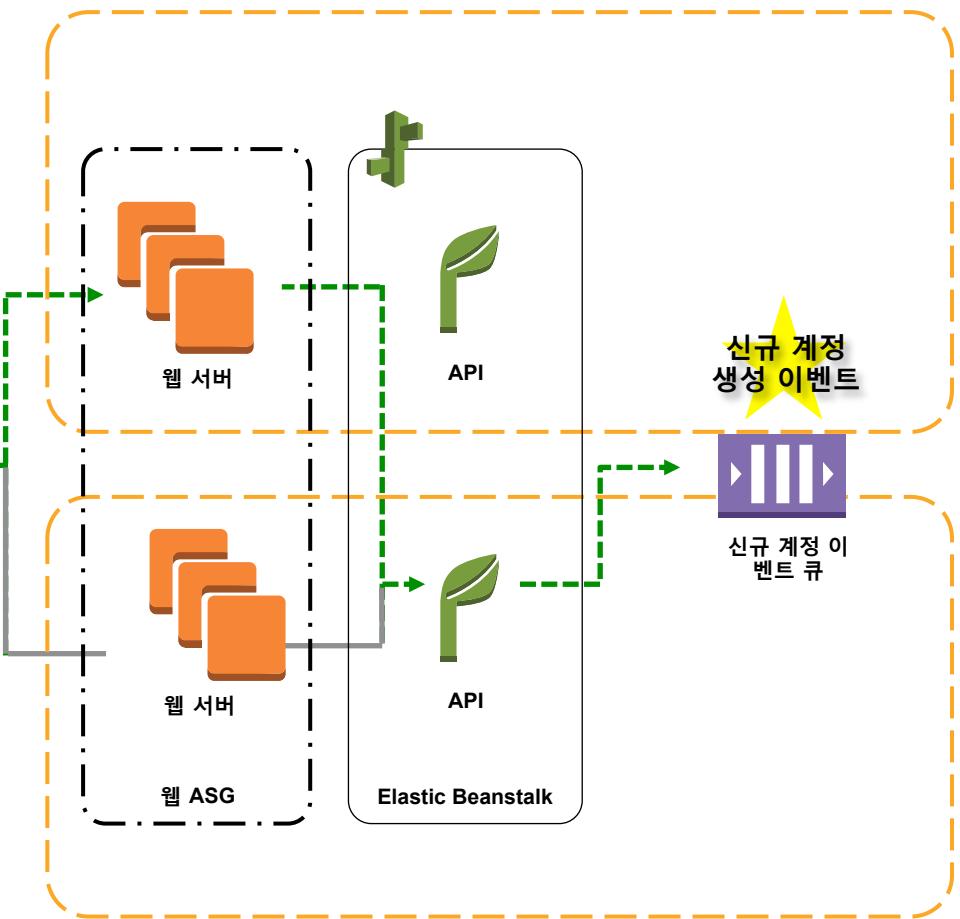
신규 계정 생성 이벤트: 사용자가 새로운 계정을 생성할 때 발생하는 이벤트입니다. 큐에 저장되는 이벤트 메시지는 사용자 이름, 암호, 홈 지역을 포함합니다.

DocStore



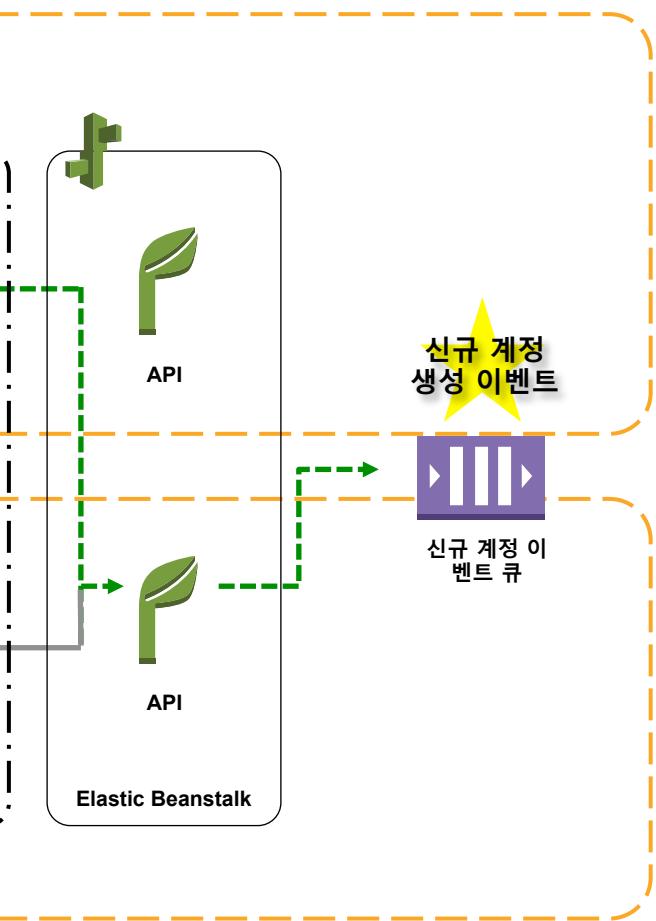
신규 계정 생성 이벤트: 사용자가 새로운 계정을 생성할 때 발생하는 이벤트입니다. 큐에 저장되는 이벤트 메시지는 사용자 이름, 암호, 홈 지역을 포함합니다.

DocStore



신규 계정 생성 이벤트: 사용자가 새로운 계정을 생성할 때 발생하는 이벤트입니다. 큐에 저장되는 이벤트 메시지는 사용자 이름, 암호, 홈 지역을 포함합니다.

DocStore



신규 계정 생성 이벤트: 사용자가 새로운 계정을 생성할 때 발생하는 이벤트입니다. 큐에 저장되는 이벤트 메시지는 사용자 이름, 암호, 홈 지역을 포함합니다.

DocStore

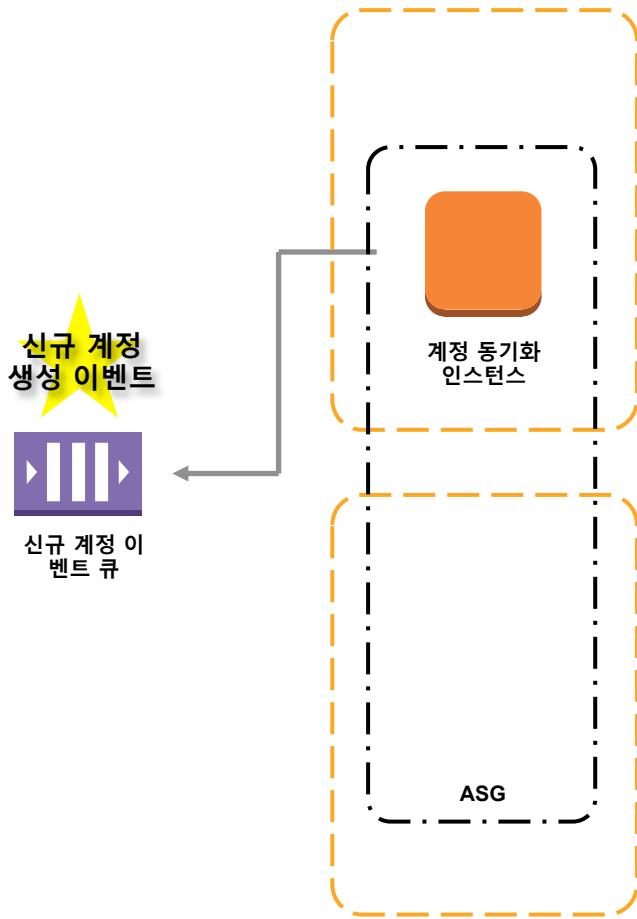


신규 계정
이벤트 큐

 amazon
web services™

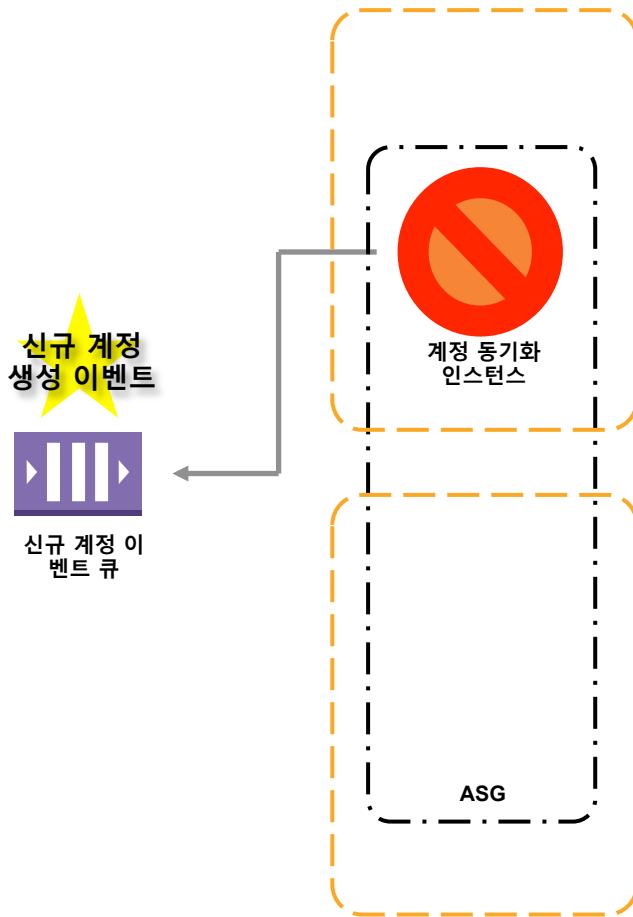
신규 계정 생성 이벤트: 계정 동기화 서비스를 구동하는 인스턴스는 신규 계정 이벤트 큐를 폴링합니다.

DocStore



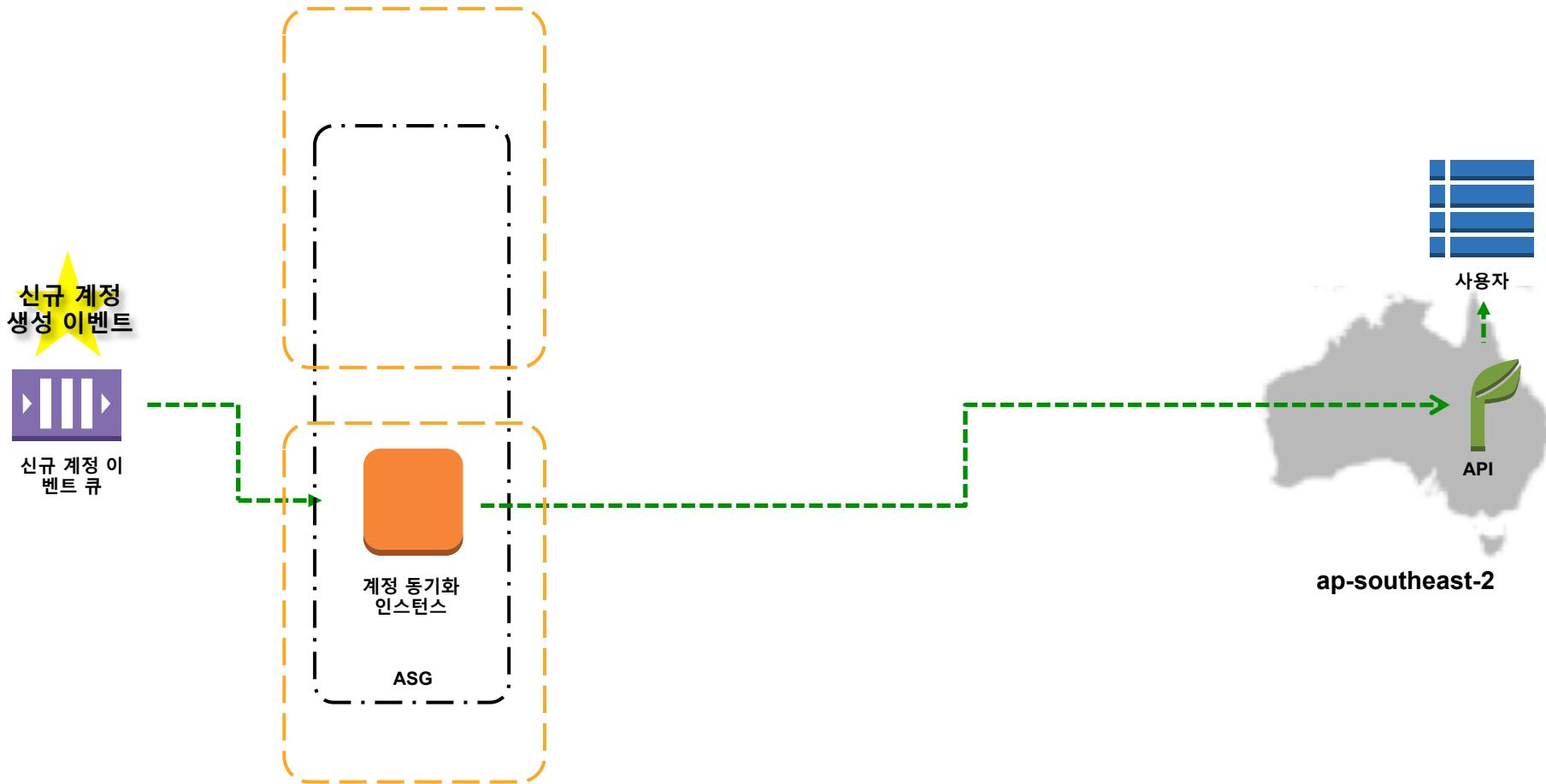
신규 계정 생성 이벤트: 인스턴스에 장애가 발생하면 Auto Scaling은 대체 인스턴스를 자동으로 생성합니다.

DocStore



신규 계정 생성 이벤트: 계정 동기화 인스턴스는 ap-southeast-2의 API를 호출하여 새로운 사용자의 계정 데이터를 동기화합니다.

DocStore



문서 업로드 시작 이벤트: 사용자가 S3로 문서 업로드를 시작하면 이벤트가 실행되어 워크플로우를 통해 업로드를 추적할 수 있습니다.

DocStore

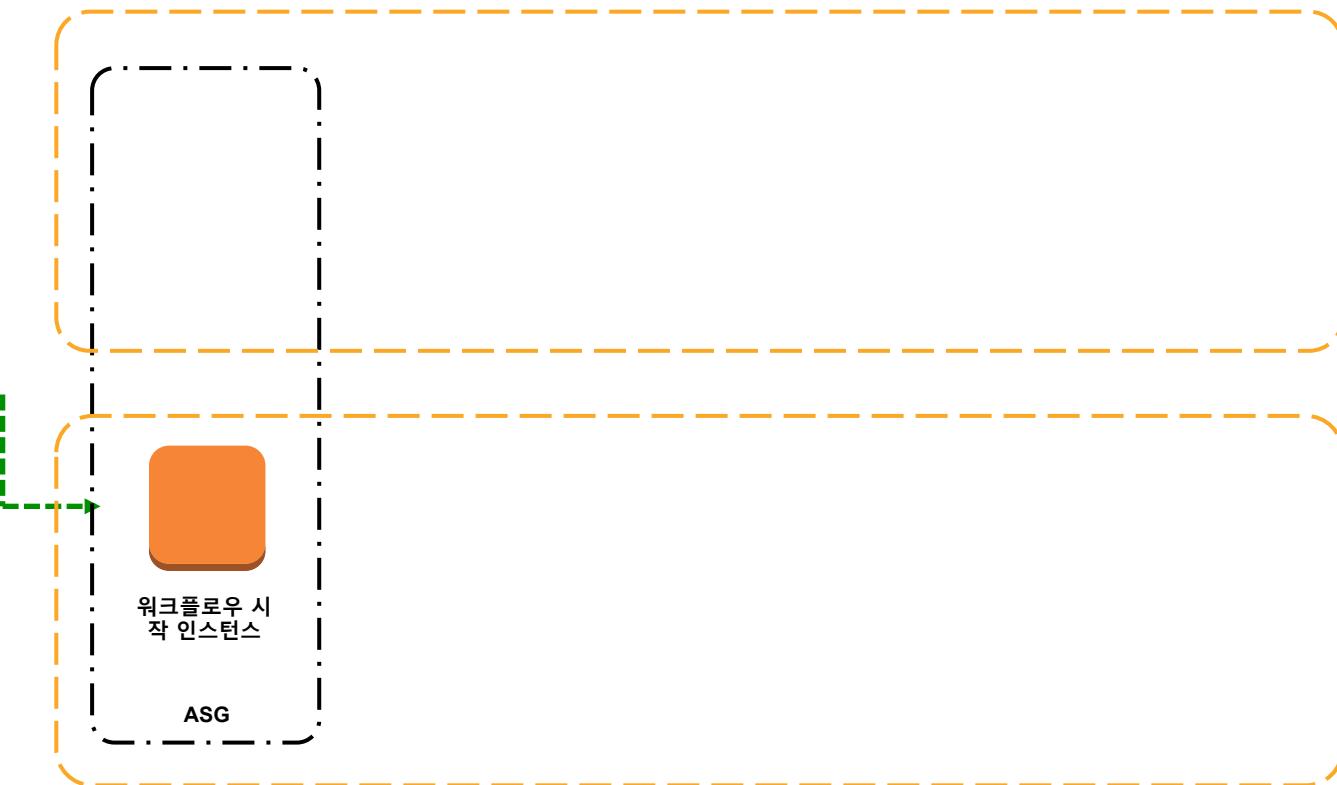


문서 업로드
시작

 amazon
web services™

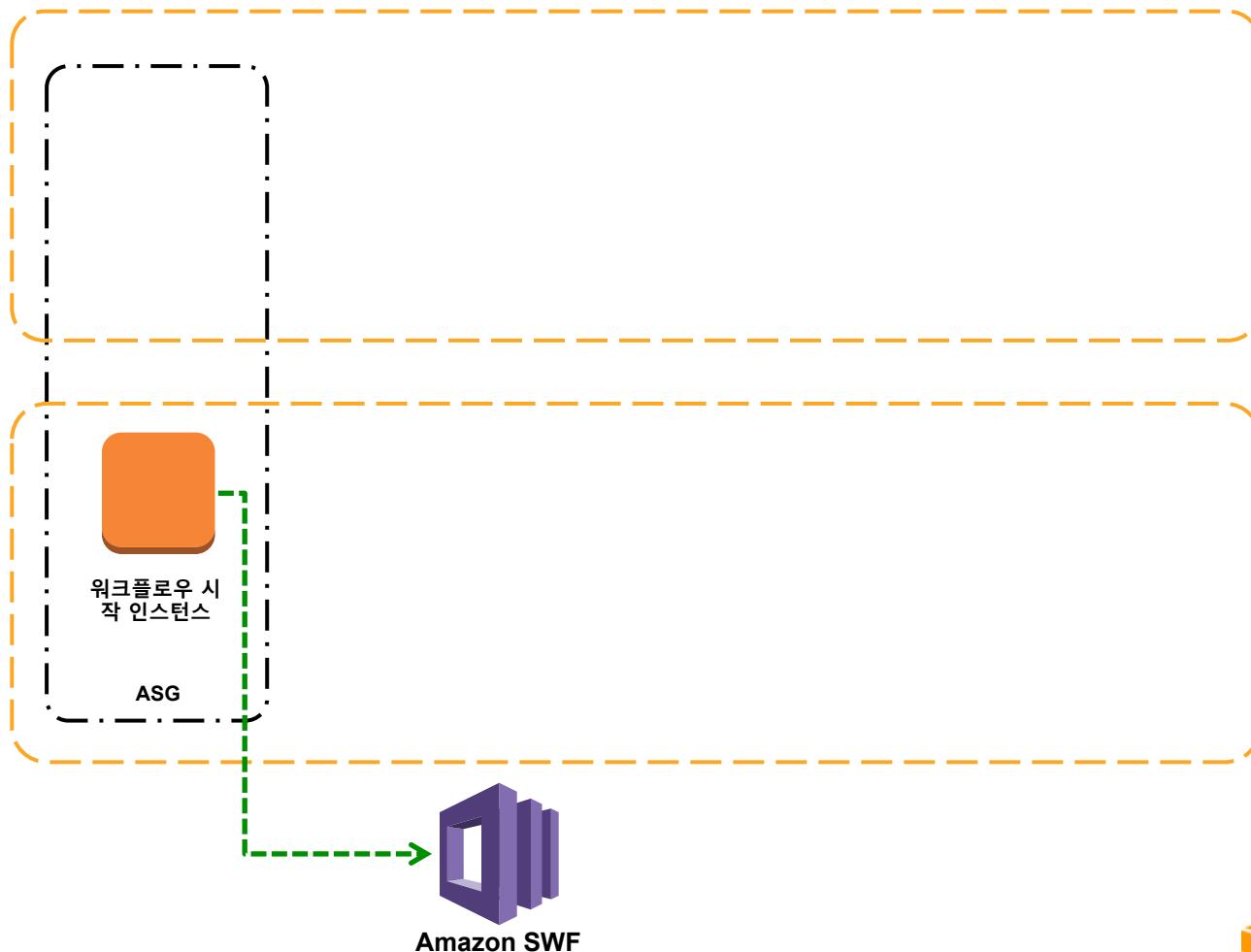
문서 업로드 시작 이벤트: 큐로부터 이벤트가 수신되면 워크플로우 시작 인스턴스는 SWF 워크플로우를 시작합니다.

DocStore



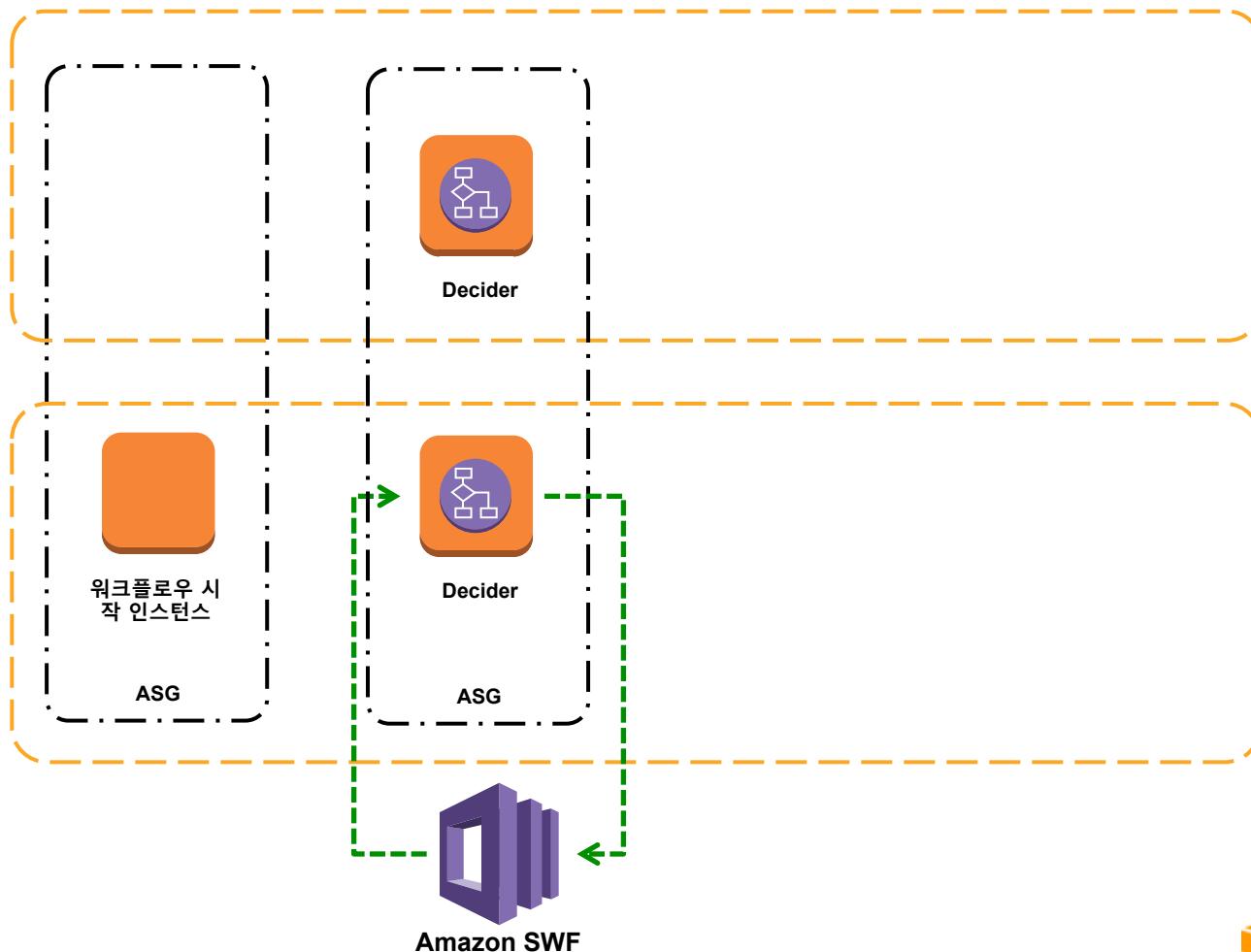
문서 업로드 시작 이벤트: 큐로부터 이벤트가 수신되면 워크플로우 시작 인스턴스는 SWF 워크플로우를 시작합니다.

DocStore



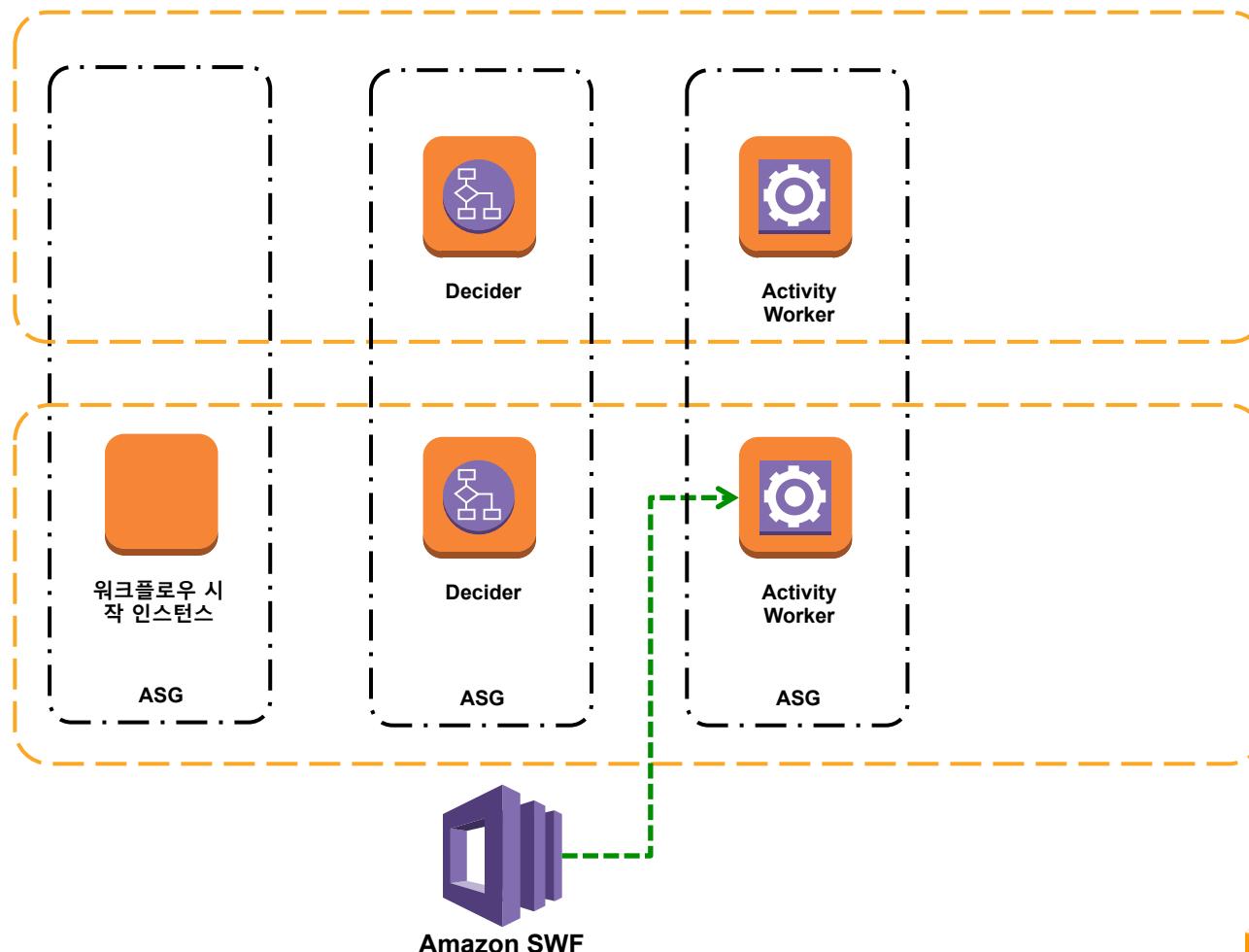
문서 업로드 시작 이벤트: Decider 코드를 구동하는 인스턴스는 워크플로우 실행 작업을 수신해, 업로드 상태 확인 활동을 예약합니다.

DocStore



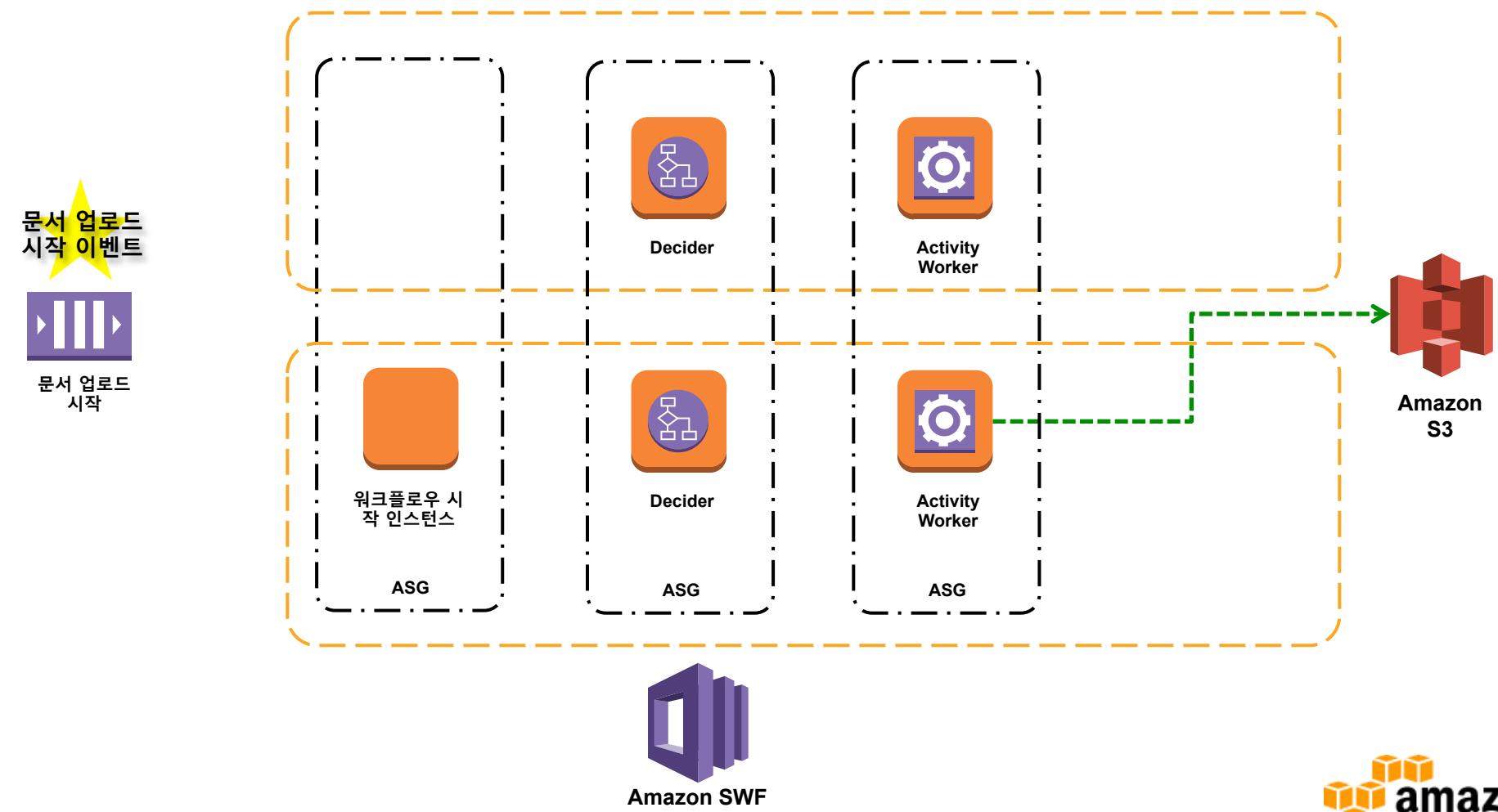
문서 업로드 시작 이벤트: Activity Worker를 구동하는 인스턴스는 업로드 상태 확인 작업을 수행합니다.

DocStore



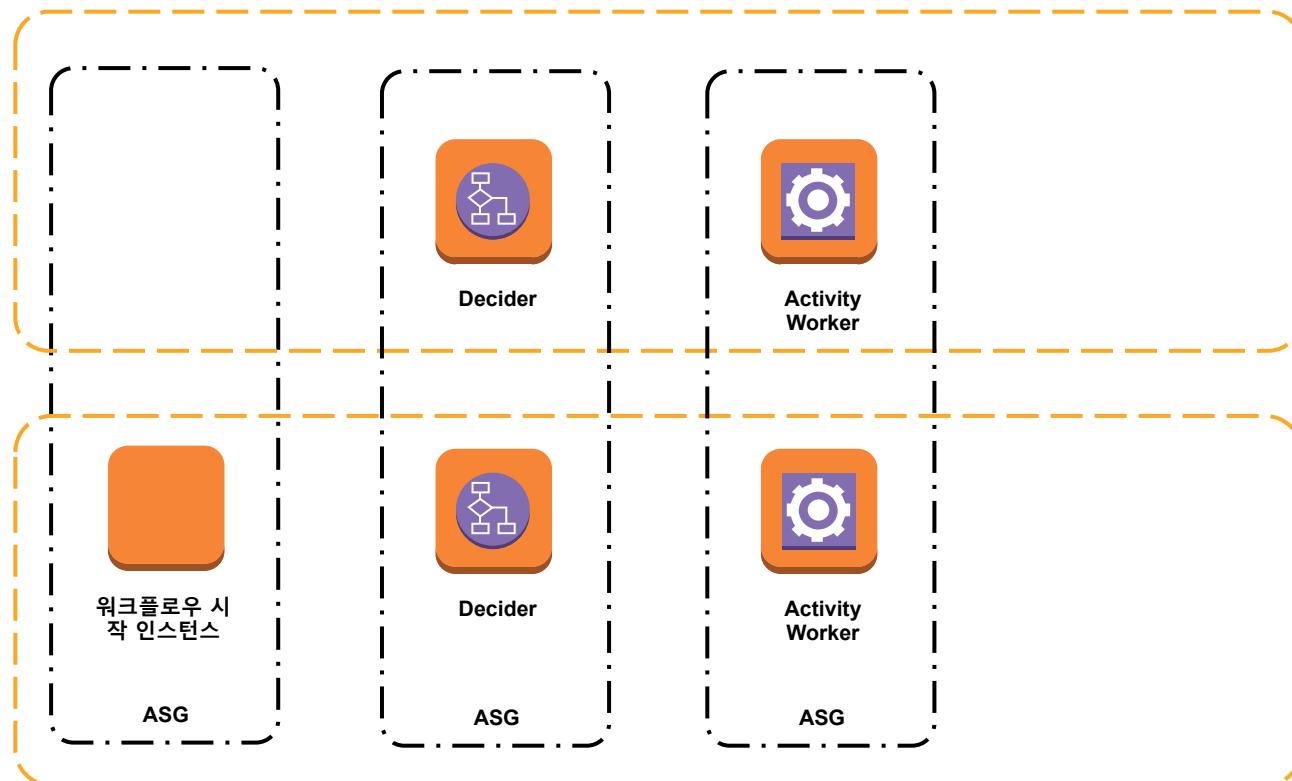
문서 업로드 시작 이벤트: Activity Worker는 주기적으로 S3를 확인하여 문서 업로드가 완료되었는지(즉, 객체가 S3 내에 존재하는지) 확인합니다.

DocStore



문서 업로드 시작 이벤트: S3상의 문서가 일단 확인되면 **문서 업로드 완료** 워크플로우에 의해 문서가 제대로 처리되었는지 확인하는 마지막 작업이 예약됩니다.

DocStore



문서 업로드 완료 이벤트: 문서 업로드가 완료되면 추가 작업
이 필요합니다.

DocStore

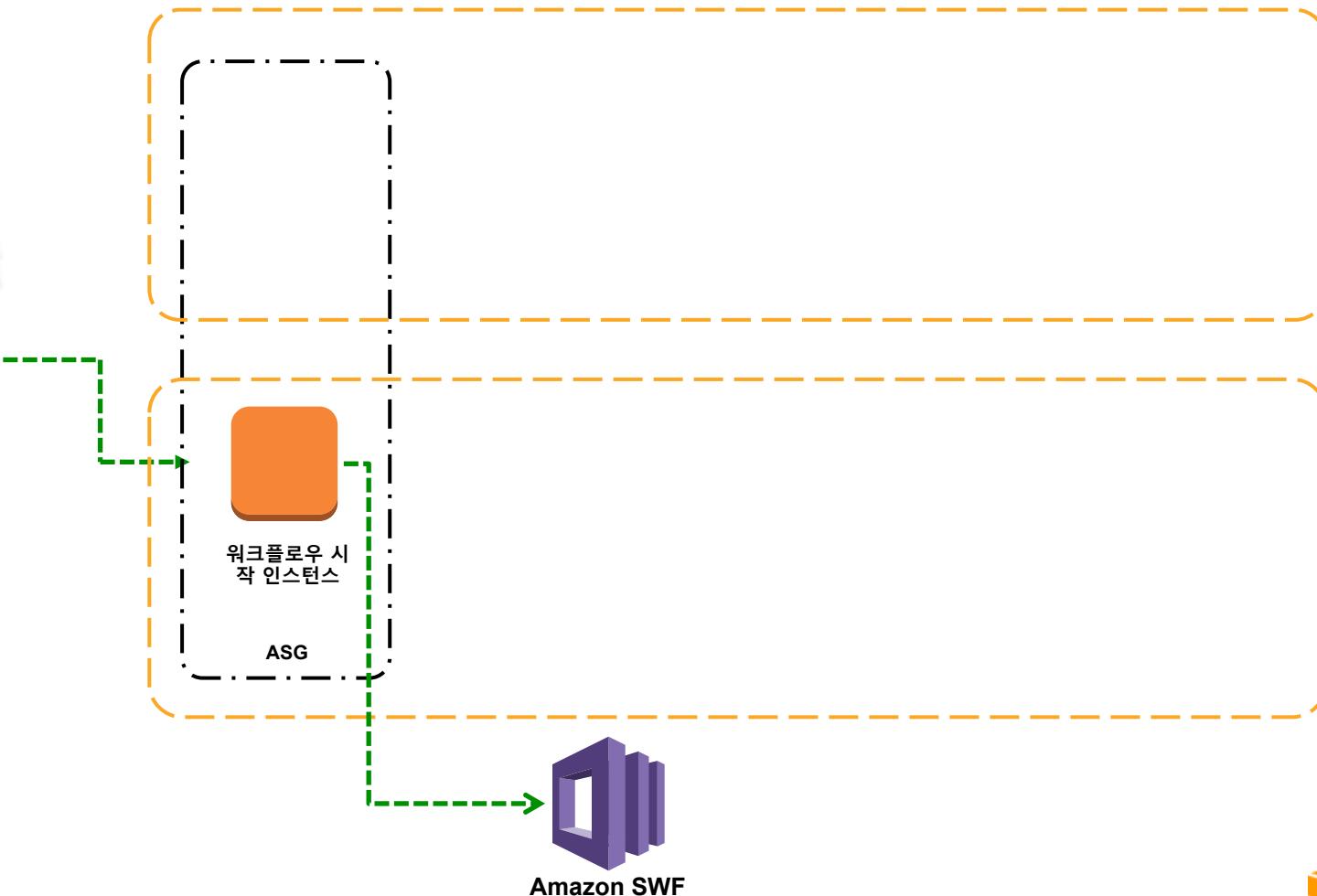


문서 업로드
완료



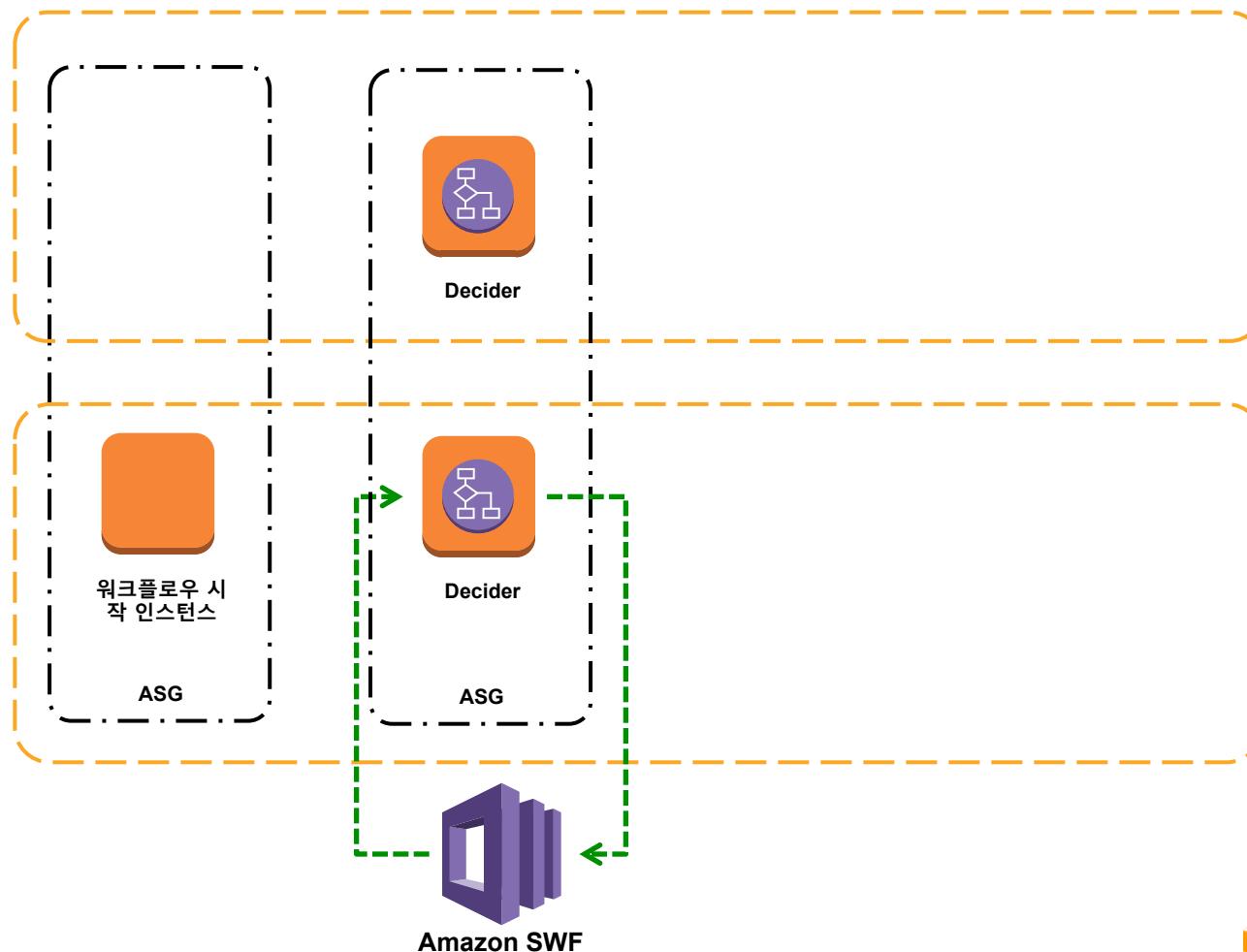
문서 업로드 완료 이벤트: 워크플로우 시작 인스턴스는 고객 유형, 문서 유형 등에 기초하여 적합한 워크플로우를 결정해 시작합니다.

DocStore



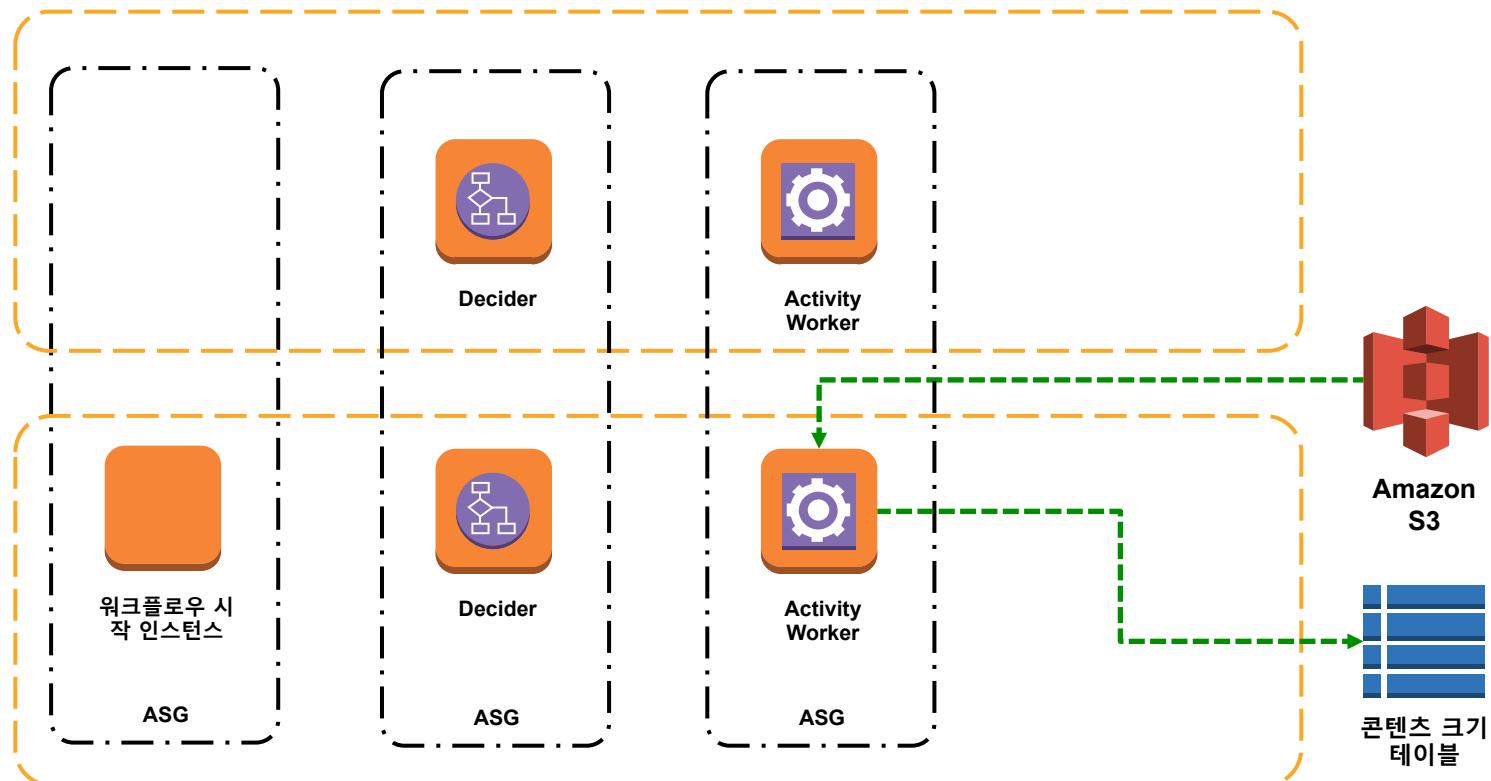
문서 업로드 완료 이벤트: Decider는 문서 유형에 따라 적절한 작업을 예약합니다.

DocStore

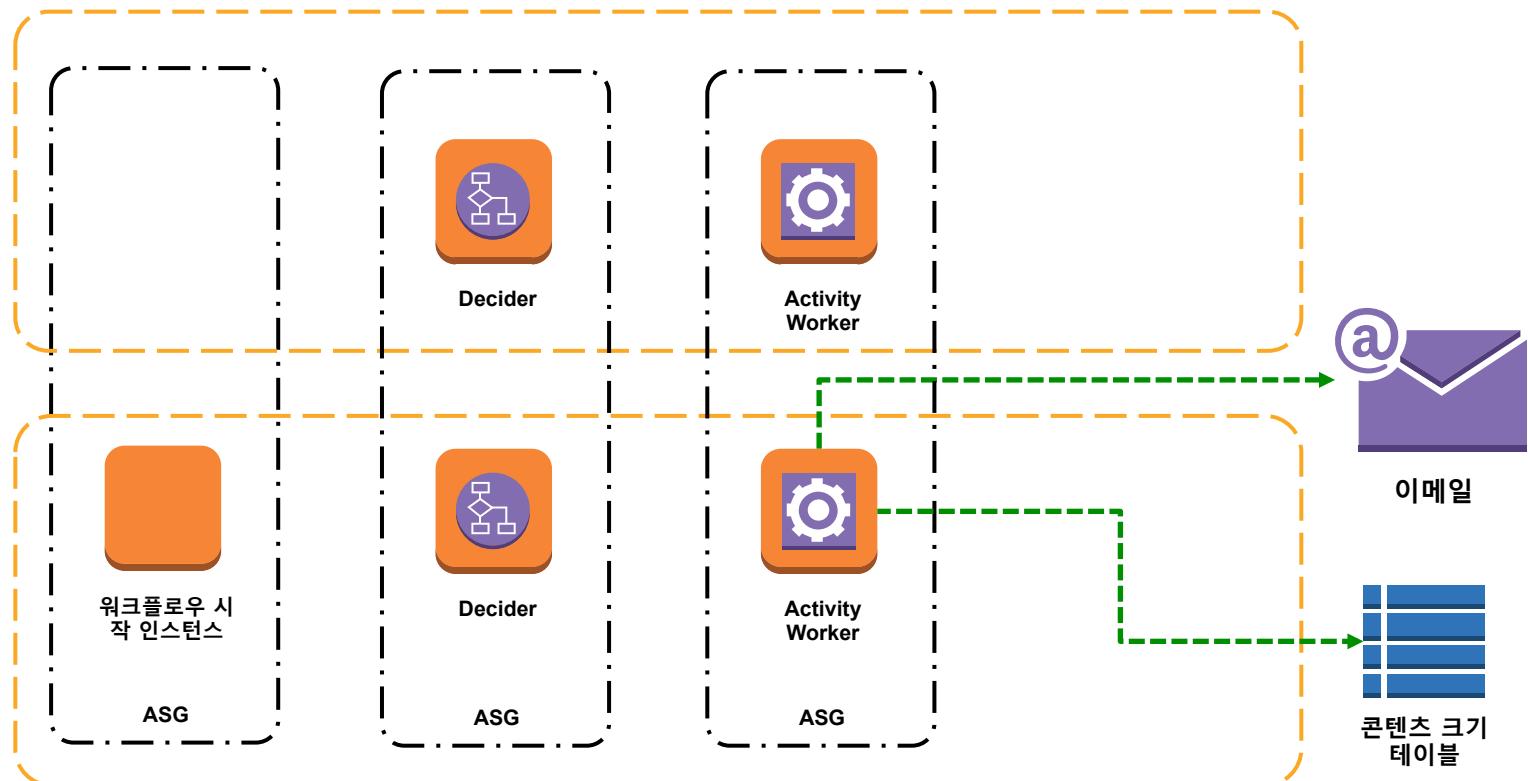


문서 업로드 완료 이벤트: 모든 객체의 **content-length** 메타데이터 속성은 S3로부터 조회되어 업로드된 모든 객체의 총량을 추적하는 카운터를 업데이트하는 데 사용됩니다.

DocStore

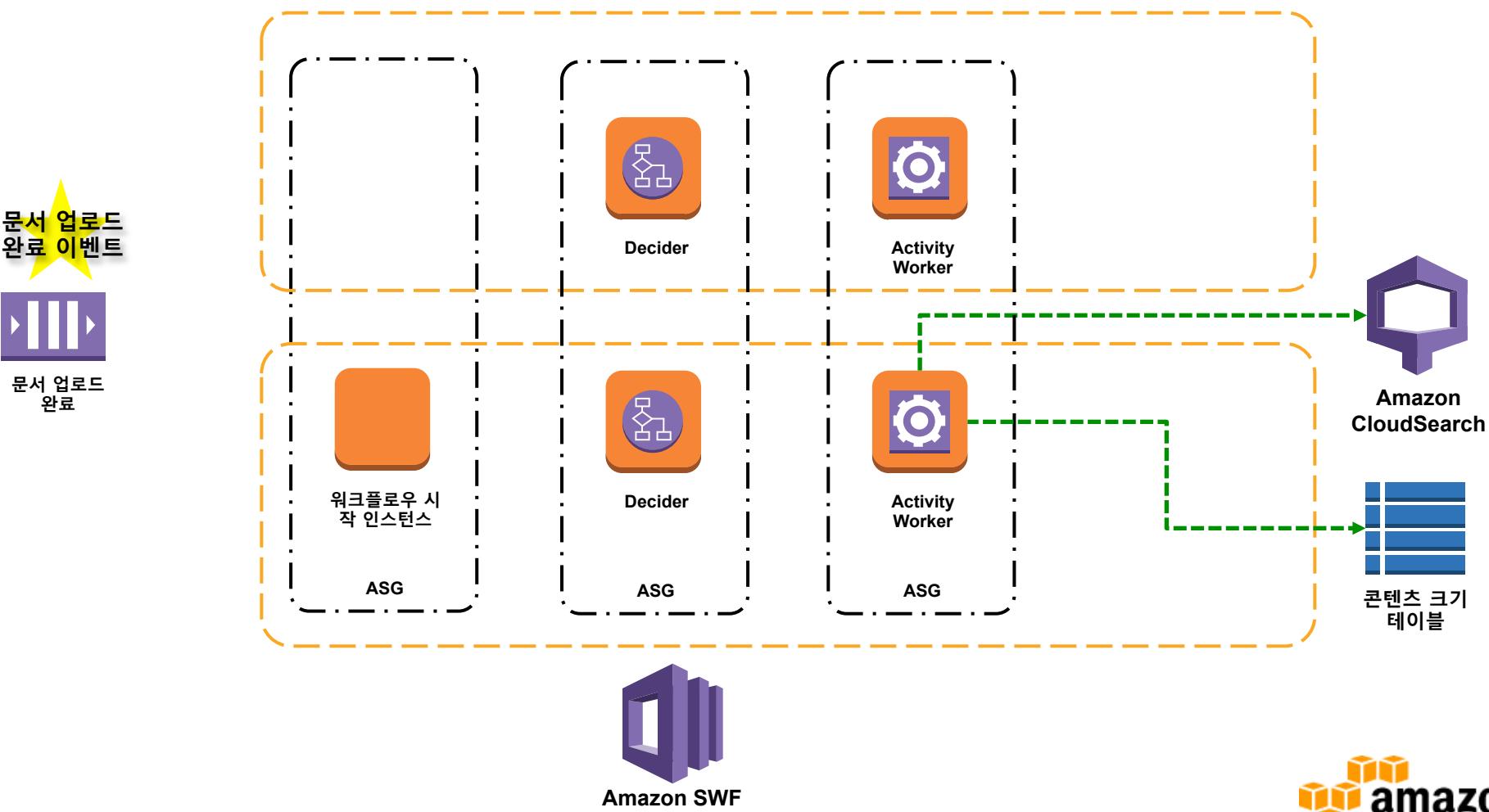


문서 업로드 완료 이벤트: 사용자별 업로드 콘텐츠 총량을 추적하는 카운터도 업데이트됩니다. 새로 업로드된 문서로 인해 해당 사용자의 할당량이 초과되면 플래그를 설정하고 **SES**를 통해 사용자에게 이메일을 발송합니다.



문서 업로드 완료 이벤트: 문서가 프리미엄 사용자에 의해 업로드되었고 PDF, DOCX와 같이 내용 검색이 가능한 문서 유형이면 **CloudSearch**를 통해 인덱싱됩니다.

DocStore



문서 업로드 완료 이벤트:마지막으로, 여러 사용자 지정 메트릭이 CloudWatch로 업데이트되어 업로드된 신규 문서 개수, 문서 누적 크기 등을 사용자 유형별로 모니터링하는 데 사용됩니다.

