

EC2 인스턴스 원격 재부팅

NULL4U 2학년 이희찬

목차

01. 주제 선정 이유

02. 사용한 서비스 소개

03. 구현 과정

04. 시행 영상

주제 선정 이유



최정훈 선생님

“학과 사이트가 가끔 다운될 때가 있다”

주제 선정 이유



최정훈 선생님

“그때마다 일일이 AWS 로그인하고 재부팅 놀러야 한다”

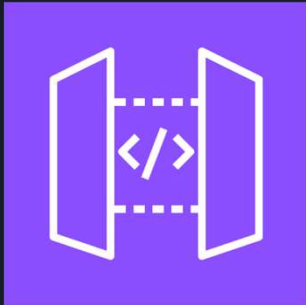
주제 선정 이유



최정훈 선생님

“원격으로 포털 수동 재부팅 가능할 것 같은데?”

서비스 설명

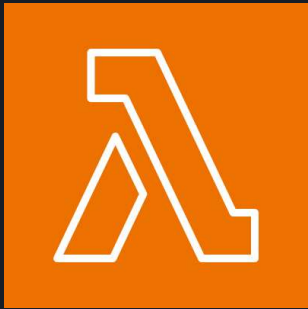


API Gateway

API 생성, 게시, 유지 관리 등을 할 수 있게 해주는 서비스

사용자가 요청을 보낼 때 이를 받아서 Lambda 함수로 전달하고,
응답을 다시 사용자에게 보내주는 중간 다리 역할

서비스 설명



Lambda

서버를 관리할 필요 없이 코드를 실행하게 해주는 서버리스 컴퓨팅 서비스

API Gateway로부터 요청이 들어오면

Lambda에서 Python 코드를 실행해, EC2 인스턴스를 재부팅

서비스 설명



IAM

AWS 리소스에 대한 접근을 관리하고 제어할 수 있는 서비스

Lambda가 EC2 인스턴스를 재부팅하려면 EC2를 제어할 권한이 필요한데, 그 권한을 부여하는 역할

작동 원리



최정훈 선생님



데스크탑, 모바일 브라우저에서
API 호출

작동 원리



최정훈 선생님



<https://api-id.execute-api.region.amazonaws.com/stage/>



API Gateway

1. HTTP 요청 받음
2. API 키 검증
3. 유효 -> Lambda로 전달



AWS Lambda

1. HTTP 요청 받음
2. EC2 인스턴스 재부팅

작동 원리



최정훈 선생님



<https://api-id.execute-api.region.amazonaws.com/stage/>



API Gateway

1. HTTP 요청 받음
2. API 키 검증
3. 유효 -> Lambda로 전달



AWS Lambda

1. HTTP 요청 받음
2. EC2 인스턴스 재부팅

Lambda 함수

함수 생성 정보

다음 옵션 중 하나를 선택하여 함수를 생성합니다.

새로 작성

간단한 Hello World 예제는 시작하십시오.

기본 정보

함수 이름

함수의 용도를 설명하는 이름을 입력합니다.

RebootInstance

함수 이름은 1~64자여야 하고, 리전에 고유해야 하며, 공백을 포함

런타임 정보

Choose the language to use to write your function. Note that th

Python 3.13

아키텍처 정보

함수 코드에 대해 원하는 명령 세트 아키텍처를 선택합니다.

x86_64

arm64

코드 소스 정보

📘 You are using the new console editor.

```
lambda_function.py X
lambda_function.py
1 import boto3
2 import json
3
4 def lambda_handler(event, context):
5     ec2 = boto3.client('ec2', region_name='ap-northeast-2')
6     instance_id = 'i-09dc8d08979526b68' # null3u.net 인스턴스 ID
7
8     try:
9         ec2.reboot_instances(InstanceIds=[instance_id])
10        return {
11            'statusCode': 200,
12            'body': json.dumps(f'Instance {instance_id} rebooted successfully!')
13        }
14    except Exception as e:
15        return {
16            'statusCode': 500,
17            'body': json.dumps(str(e))
18        }
19
```

boto3 : 파이썬에서 aws 제어할수 있게 해주는 모듈

IAM 권한 생성

정책 세부 정보

정책 이름

이 정책을 식별하는 의미 있는 이름을 입력합니다.

Allow_EC2_Reboot

최대 128자입니다. 영숫자 및 '+', '@', '-' 문자를 사용

권한 지정 정보

서비스, 작업, 리소스 및 조건을 선택하여 권한을 추가합니다. JSON 편집기를 사용하여 권한 설명문을 작성합니다.

정책 편집기

시각적

JSON

```
1 |  
2 | "Version": "2012-10-17",  
3 | "Statement": [  
4 |   {  
5 |     "Effect": "Allow",  
6 |     "Action": "ec2:RebootInstances",  
7 |     "Resource": "arn:aws:ec2:ap-northeast-2:172020907737:instance/i-09dc8d08979526b68" # Null3u.net의 인스턴스 id  
8 |   }  
9 | ]  
10 |
```

문 편집

IAM 권한 생성

🕒 정책 Allow_EC2_Reboot이(가) 생성되었습니다.

정책 (1320) [정보](#)

정책은 권한을 정의하는 AWS의 객체입니다.

🔍 allow

| 정책 이름

[Allow_EC2_Reboot](#)

Lambda와 IAM 연결

RebootInstance-role-ucgoksnt 정보

Lambda 기본 권한 정책

[삭제](#)

요약

[편집](#)

생성 날짜
December 31, 2024, 11:42 (UTC+09:00)

ARN
arn:aws:iam::172020907737:role/service-role/RebootInstance-role-ucgoksnt

마지막 활동
5시간 전

최대 세션 지속 시간
1시간

[권한](#) | [신뢰 관계](#) | [태그](#) | [마지막 액세스](#) | [세션 취소](#)

권한 정책 (1) 정보

최대 10개의 관리형 정책을 연결할 수 있습니다.

[시뮬레이션](#) | [삭제](#) | [권한 추가](#)

[정책 연결](#)
[인라인 정책 생성](#)

검색	필터링 기준 유형	연결된 엔터티
<input type="checkbox"/> 정책 이름	모든 유형	
<input type="checkbox"/> AWSLambdaBasicExecutionRole-d85f1f58-c411-4608-97df-17...	고객 관리형	1

기타 권한 정책 (1/1024)

검색	필터링 기준 유형	연결된 엔터티	설명
<input checked="" type="checkbox"/> Allow_EC2_Reboot	고객 관리형		
<input type="checkbox"/> AWSIAMIdentityCenterAllowListForIdentityContext	AWS 관리형		Provides the list of actions that are all...

[취소](#)[권한 추가](#)

Lambda 테스트

The screenshot displays the AWS Lambda console interface. On the left, the 'EXPLORER' pane shows the file structure for the 'REBOOTINSTANCE' function, with 'lambda_function.py' selected. Below this, the 'DEPLOY' section contains two buttons: 'Deploy (Ctrl+Shift+U)' and 'Test (Ctrl+Shift+I)', with the latter highlighted by a red border. The 'TEST EVENTS [SELECTED: REBOOTTEST]' section shows a single event named 'RebootTest'.

The main editor area shows the Python code for 'lambda_function.py':

```
1 import boto3
2 import json
3
4 def lambda_handler(event, context):
5     ec2 = boto3.client('ec2', region_name='ap-northeast-2')
6     instance_id = 'i-09dc8d08979526b68' # null3u.net 인스턴스 ID
7
8     try:
9         ec2.reboot_instances(InstanceIds=[instance_id])
10        return {
11            'statusCode': 200,
12            'body': json.dumps(f'Instance {instance_id} rebooted successfully!')
13        }
14    except Exception as e:
15        return {
16            'statusCode': 500,
17            'body': json.dumps(str(e))
18        }
```

PROBLEMS OUTPUT CODE REFERENCE LOG TERMINAL

Status: Succeeded
Test Event Name: RebootTest

Response:

```
{
  "statusCode": 200,
  "body": "\"Instance i-09dc8d08979526b68 rebooted successfully!\""
}
```


작동 원리



최정훈 선생님



<https://api-id.execute-api.region.amazonaws.com/stage/>



API Gateway

1. HTTP 요청 받음
2. API 키 검증
3. 유효 -> Lambda로 전달



done



AWS Lambda

1. HTTP 요청 받음
2. EC2 인스턴스 재부팅

작동 원리



최정훈 선생님



<https://api-id.execute-api.region.amazonaws.com/stage/>



API Gateway

1. HTTP 요청 받음
2. API 키 검증
3. 유효 -> Lambda로 전달



done



AWS Lambda

1. HTTP 요청 받음
2. EC2 인스턴스 재부팅

API 생성

API Gateway 서비스로 이동

REST API 생성

API 세부 정보

새 API
새 REST API를 생성합니다.

API 가져오기
OpenAPI 정의에서 API를 가져옵니다.

API 이름

RebootEC2API

설명 - 선택 사항

API 엔드포인트 유형

리전 API는 현재 AWS 리전에 배포되어 있습니다. 엣지 최적화 API는 요청을 가장 가까운 CloudFront 접속 지점으로 라우팅합니다. 프라이빗 API는 VP

지역

API 생성

Successfully created REST API 'RebootEC2API (ony39it269)'. ✕

리소스

API 작업 ▾

API 배포

리소스 생성

리소스 세부 정보

설명서 업데이트

CORS 활성화

경로

/

리소스 ID

t19c634ym4

메서드 (0)

삭제

메서드 생성

메서드 유형 ▲

통합 유형 ▼

권한 부여 ▼

API 키 ▼

메서드 없음

정의된 메서드가 없습니다.

API 생성

리소스 생성

리소스 세부 정보

- 프록시 리소스 정보**
프록시 리소스는 모든 하위 리소스에 대한 요청을 처리합니다. 프록시 리소스를 생성하려면 더하기 기호로 끝나는 경로 파라미터(예: {proxy+})를 사용합니다.

리소스 경로

/

리소스 이름

reboot

- 오리진 간 리소스 공유(CORS) 정보**
모든 오리진, 모든 메서드 및 몇 가지 공통 헤더를 허용하는 OPTIONS 메시지를 생성합니다.

취소

리소스 생성

API 생성

메서드 생성

메서드 세부 정보

메서드 유형

POST

통합 유형

Lambda 함수
API를 Lambda 함수와 통합합니다.



AWS 서비스
AWS 서비스와 통합합니다.



Lambda 프록시 통합
요청을 구조화된 이벤트로 Lambda 함수에 전송합니다.

Lambda 함수

Lambda 함수 이름 또는 별칭을 제공합니다. 다른 계정의 ARN을 제공할 수도 있습니다.

ap-northeast-2

arn:aws:lambda:ap-northeast-2:172020907737:function:RebootInsta X

API Gateway에 Lambda 함수를 간접적으로 호출할 권한을 부여합니다. 이 기능을 끄려면 함수의 리소스 정책을 직접 업데이트

통합 시간 초과 정보

기본적으로 통합 제한 시간을 50밀리초에서 2만 9,000밀리초로 입력할 수 있습니다. Service Quotas를 사용하여 통합 제한 시간을 2만 9,000밀리초

29000

API 생성

✔ 'reboot'에서 메서드 'POST'을(를) 생성했습니다. 업데이트를 적용하려면 API를 재배포하세요.



리소스

API 작업 ▾

API 배포

리소스 생성



/reboot

POST

/reboot - POST - 메서드 실행

설명서 업데이트

삭제

ARN

am:aws:execute-api:ap-northeast-2:172020907737:ony39it269/*/POST/reboot

리소스 ID

7mh8eb



클라이언트



메서드 요청



통합 요청



Lambda
통합



메서드 응답



통합 응답



메서드 요청

통합 요청

통합 응답

메서드 응답

테스트

메서드 요청서지

편집

API 생성

/reboot - POST - 메서드 실행

Deploy API



API를 배포할 스테이지를 생성하거나 선택합니다. 배포 기록을 사용하여 스테이지의 활성 배포를 되돌리거나 변경할 수 있습니다. [Learn more](#)

스테이지

새 스테이지



스테이지 이름

prod

i 기본 설정으로 새 스테이지가 생성됩니다. 스테이지 페이지에서 스테이지 설정을 편집합니다.

배포 설명

취소

배포

API 생성

prod
/
/reboot
POST

메서드 재정의

기본적으로 메서드는 스테이지 수준 설정을 상속합니다. 메서드의 설정을 사용자 지정하

i 이 메서드는 'prod' 스테이지의 설정을 상속합니다.

URL 호출

<https://ony39it269.execute-api.ap-northeast-2.amazonaws.com/prod/reboot>

API 호출할 수 있는 URL

작동 원리



최정훈 선생님



<https://api-id.execute-api.region.amazonaws.com/stage/>



API Gateway

1. HTTP 요청 받음
2. API 키 검증
3. 유효 -> Lambda로 전달



done



AWS Lambda

1. HTTP 요청 받음
2. EC2 인스턴스 재부팅

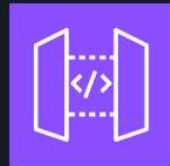
작동 원리



최정훈 선생님



<https://api-id.execute-api.region.amazonaws.com/stage/>



API Gateway

1. HTTP 요청 받음
2. API 키 검증
3. 유효 -> Lambda로 전달

done



AWS Lambda

1. HTTP 요청 받음
2. EC2 인스턴스 재부팅

작동 원리



최정훈 선생님

done



<https://ony39it269.execute-api.ap-northeast-2.amazonaws.com/prod/reboot>



API Gateway

1. HTTP 요청 받음
2. API 키 검증
3. 유효 -> Lambda로 전달

done



AWS Lambda

1. HTTP 요청 받음
2. EC2 인스턴스 재부팅

API 키 생성

≡ [API Gateway](#) > [API](#) > [API 키](#) > API 키 생성

API 키 생성 정보

API 키 세부 정보

이름

Rebootkey

설명 - 선택 사항

API 키

- 자동 생성
 사용자 지정

[취소](#)

[저장](#)

API 생성

RebootKey

API 키 세부 정보

ID

hm48v4g3q8

설명

-

생성 날짜

January 02, 2025, 11:34 (UTC+09:00)

상태

✔ 활성화

API 키

□ Show

이 키를 사용량 계획과 연동해 특정 클라이언트만 API를 호출할 수 있도록 설정해줄 것

사용량 계획이란?

클라이언트가 요청을 몇 번이나, 얼마나 자주 보낼 수 있는지 제어하는 기능

사용량 계획 생성

사용량 계획 생성 정보

사용량 계획 세부 정보

사용량 계획은 API가 클라이언트로부터 수락하는 요청의 수 또는 비율을 지정합니다. A

이름

RebootPlan

설명 - 선택 사항

조절 중

사용자가 API를 직접적으로 호출할 수 있는 속도를 제한합니다.

요율

클라이언트가 API를 직접적으로 호출할 수 있는 속도를 초당 요청 수로 입력합니다.

10

버스트

클라이언트가 API에 요청할 수 있는 동시 요청 수를 입력합니다.

1

할당량

할당량을 활성화하여 사용자 한 명이 특정 기간 동안 API에 보낼 수 있는 요청 수를 제한합니다.

사용량 계획 생성

RebootPlan

작업 ▾

사용량 데이터 내보내기

사용량 계획 세부 정보

사용량 계획 ID
vr7ax3

설명
-

AWS Marketplace 제품 코드
-

요율
초당 요청 10건

버스트
1건의 요청

할당량
-

연결된 스테이지

연결된 API 키

태그

연결된 스테이지 (0) [정보](#)

편집

제거

스테이지 추가

API



스테이지



메서드 제한

스테이지 없음

스테이지가 없습니다.

API 스테이지 추가

사용량 계획 생성

스태이지 연결 정보

스태이지 세부 정보

API

RebootEC2API

스태이지

prod

스태이지를 선택하려면 먼저 API를 선택해야 합니다.

▶ 메서드 수준 제한 - 선택 사항

취소

사용량 계획에 추가

사용량 계획 생성

RebootPlan

작업 ▼

사용량 데이터 내보내기

사용량 계획 세부 정보

사용량 계획 ID
vr7ax3

설명
-

AWS Marketplace 제품 코드
-

요율
초당 요청 10건

버스트
1건의 요청

할당량
-

연결된 스테이지

연결된 API 키

태그

연결된 스테이지 (1) 정보

편집

제거

스테이지 추가

API

▼

스테이지

▼

메서드 제한

○

[RebootEC2API](#)

prod

-

사용량 계획 생성

RebootPlan

작업 ▼

사용량 데이터 내보내기

사용량 계획 세부 정보

사용량 계획 ID
vr7ax3

설명
-

AWS Marketplace 제품 코드
-

요율
초당 요청 10건

버스트
1건의 요청

할당량
-

연결된 스테이지

연결된 API 키

태그

API 키 (0) 정보

작업 ▼

API 키 추가

< 1 > ⚙

이름 ▼

상태 ▼

ID ▼

API 키 ▼

이번 달에 남은 요청 ▼

API 키가 없습니다.

이 사용량 계획에는 API 키가 없습니다.

API 키 추가

사용량 계획 생성

API 키 추가

API 키 세부 정보

유형

- 기존 키 추가
 새 키 생성 및 추가

API 키

RebootKey

취소

API 키 추가

사용량 계획 생성

RebootPlan

작업 ▼

사용량 데이터 내보내기

사용량 계획 세부 정보

사용량 계획 ID
vr7ax3

설명
-

AWS Marketplace 제품 코드
-

요율
초당 요청 10건

버스트
1건의 요청

할당량
-

연결된 스테이지

연결된 API 키

태그

API 키 (1) 정보

작업 ▼

API 키 추가

< 1 > ⚙

이름	상태	ID	API 키	이번 달에 남은 요청
RebootKey	✔ 활성화됨	hm48v4g3q8	0

작동 원리



최정훈 선생님

done



<https://ony39it269.execute-api.ap-northeast-2.amazonaws.com/prod/reboot>



API Gateway

1. HTTP 요청 받음
2. API 키 검증
3. 유효 -> Lambda로 전달

done



AWS Lambda

1. HTTP 요청 받음
2. EC2 인스턴스 재부팅

작동 원리



최정훈 선생님

done



<https://ony39it269.execute-api.ap-northeast-2.amazonaws.com/prod/reboot>



done



API Gateway

1. HTTP 요청 받음
2. API 키 검증
3. 유효 -> Lambda로 전달



done



AWS Lambda

1. HTTP 요청 받음
2. EC2 인스턴스 재부팅

테스트

```
1 fetch('https://ony39it269.execute-api.ap-northeast-2.amazonaws.com/prod/reboot', {
2   method: 'POST',
3   headers: {
4     'X-Api-Key': 'SDAEGToV1guU1D10uMok6kEyNFV14kZ5sE0wUzki'
5   }
6 })
7 .then(response => response.json())
8 .then(data => console.log(data))
9 .catch(error => console.error('Error:', error));
```


테스트

크롬 개발자 도구(f12)의 콘솔 탭

```
> fetch('https://ony39it269.execute-api.ap-northeast-2.amazonaws.com/prod/reboot', {  
  method: 'POST',  
  headers: {  
    'X-API-Key': 'SDAEGToV1guJ1D10uMok6kEyNFV14kZ5sE0wUzki'  
  }  
})  
.then(response => response.json())  
.then(data => console.log(data))  
.catch(error => console.error('Error:', error));
```

```
✖ ▶ Error: TypeError: Failed to fetch  
  at u (bbqge6n6s1oo3m8q.js:246:474)  
  at <anonymous>:1:1
```

테스트

CORS 정책 때문

CORS란?

웹 브라우저가 다른 도메인(Origin)에서 자원을 가져오는 것을 허용하는 보안 정책

기본적으로 브라우저는 보안상의 이유로 CORS를 비활성화 해놓음

그래서 api를 호출하려고 하면 요청을 차단하는 것

CORS 활성화

리소스

API 작업 ▼

API 배포

리소스 생성

□ /

□ /reboot

POST

리소스 세부 정보

삭제

설명서 업데이트

CORS 활성화

경로
/reboot

리소스 ID
7mh8eb

메서드 (1)

삭제

메서드 생성

메서드 유형



통합 유형



권한 부여



API 키



POST

Lambda

없음

필수

CORS 활성화

CORS 활성화

CORS 설정 정보

브라우저에서 실행하는 스크립트의 요청을 허용하려면 API에 대한 CORS(cross-origin resource sharing)를 구성하세요. 구성을 저장하면 API Gateway가 기존 CORS 설정을 새 구성으로 교체합니다.

게이트웨이 응답

선택한 게이트웨이 응답에 대해 API Gateway가 CORS를 구성합니다.

기본 4XX

기본 5XX

Access-Control-Allow-Methods

OPTIONS

POST

Access-Control-Allow-Headers

선택한 게이트웨이 응답에 대해 API Gateway가 CORS를 구성합니다.

X-api-key

Access-Control-Allow-Origin

리소스에 액세스할 수 있는 오리진들 입력합니다. 와일드카드 "*"를 사용하면 모든 오리진에서 리소스에 액세스할 수 있습니다.

*

▶ 추가 설정

취소

저장

CORS 활성화

☑ Successfully enabled CORS



▼ 세부 정보

- OPTIONS 메서드가 생성되었습니다.
- OPTIONS 메서드에 응답 모델이 빈 200 메서드 응답이 추가되었습니다.
- OPTIONS 메서드에 Mock 통합이 추가되었습니다.
- OPTIONS 메서드에 200 통합 응답이 추가되었습니다.
- OPTIONS 메서드에 Access-Control-Allow-Headers, Access-Control-Allow-Methods, Access-Control-Allow-Origin 메서드 응답 헤더가 추가되었습니다.
- OPTIONS 메서드에 Access-Control-Allow-Headers, Access-Control-Allow-Methods, Access-Control-Allow-Origin 통합 응답 헤더 매핑이 추가되었습니다.
- POST 메서드에 Access-Control-Allow-Origin 메서드 응답 헤더가 추가되었습니다.
- POST 메서드에 Access-Control-Allow-Origin 통합 응답 헤더 매핑이 추가되었습니다.

스태이지

스태이지 작업 ▼

Create stage

prod

/

/reboot

OPTIONS

POST

메서드 재정의

재정의 생성

기본적으로 메서드는 스테이지 수준 설정을 상속합니다. 메서드의 설정을 사용자 지정하려면 메서드 재정의 구성하세요.

① 이 메서드는 'prod' 스테이지의 설정을 상속합니다.

URL 호출

<https://ony39it269.execute-api.ap-northeast-2.amazonaws.com/prod/reboot>

테스트

```
> fetch('https://ony39it269.execute-api.ap-northeast-2.amazonaws.com/prod/reboot', {
  method: 'POST',
  headers: {
    'X-Api-Key': 'SDAEGToVlguUIDlOuMok6kEyNFVl4kZ5sEOwUzki'
  }
})
.then(response => response.json())
.then(data => console.log(data))
.catch(error => console.error('Error:', error));
```

```
< ▶ Promise {<pending>}
  ▶ {statusCode: 200, body: '"Instance i-09d68d08979526b68 rebooted successfully!"'}
```

```
> |
```

매번 개발자 도구를 열기는 번거로움

Bookmarklet

북마크릿이란?

북마크 안에 URL이 아니라 자바스크립트 코드를 저장

북마크를 클릭하면 자바스크립트 코드가 실행됨

Bookmarklet

The screenshot shows a web browser window with the title "Bookmarklet Maker" and the URL "caiorss.github.io/bookmarklet-maker/". The page has a header "Bookmarklet Maker" and a section titled "Github Repository".

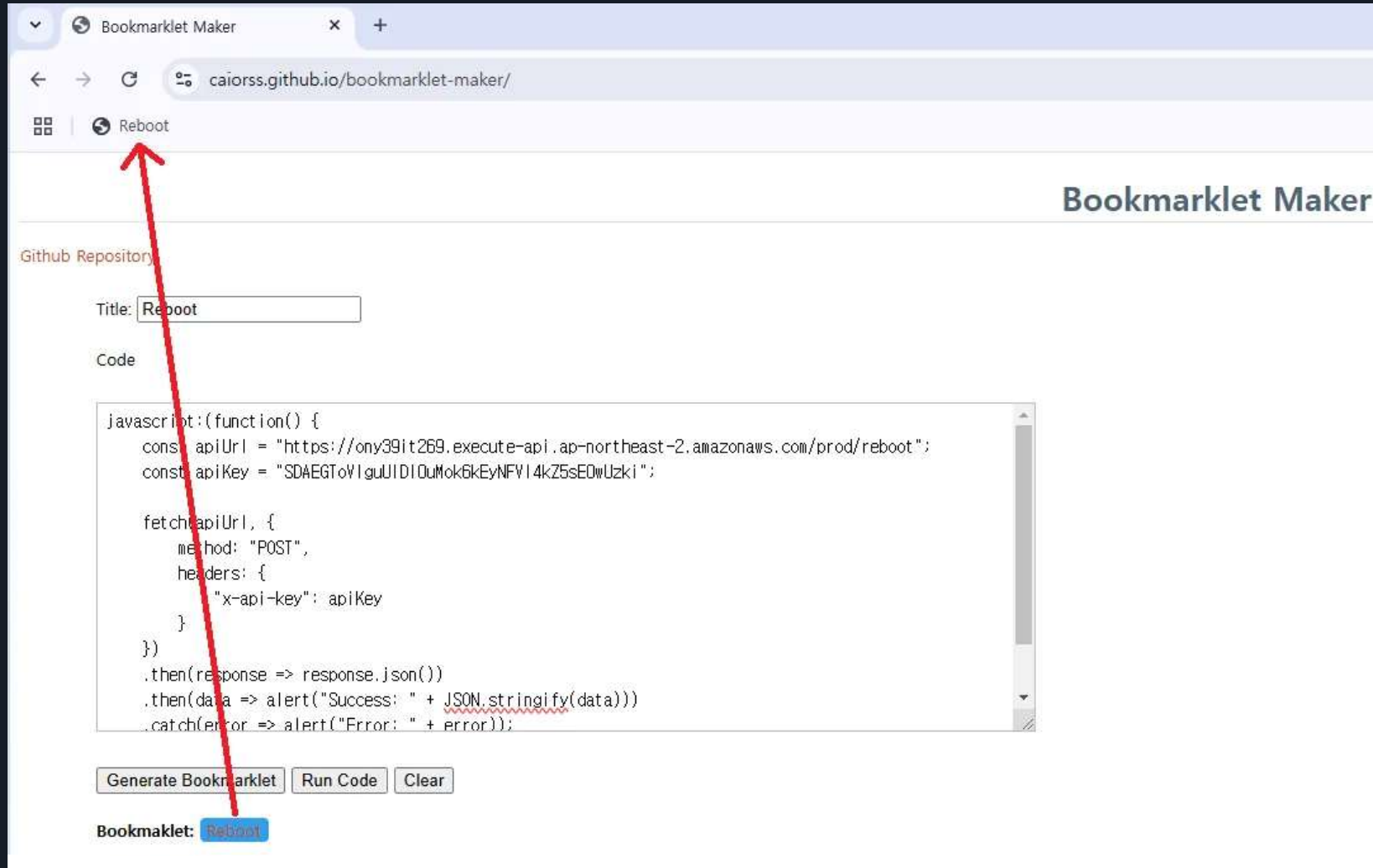
Title:

Code

```
javascript:(function() {  
  const apiUrl = "https://ony39it269.execute-api.ap-northeast-2.amazonaws.com/prod/reboot";  
  const apiKey = "SDAEGToYIguUI1D1OuMok6kEyNFYI4kZ5sEOwUzki";  
  
  fetch(apiUrl, {  
    method: "POST",  
    headers: {  
      "x-api-key": apiKey  
    }  
  })  
  .then(response => response.json())  
  .then(data => alert("Success: " + JSON.stringify(data)))  
  .catch(error => alert("Error: " + error));  
})
```

Bookmarklet: [Reboot](#)

Bookmarklet



The screenshot shows a web browser window with the address bar displaying `caiorss.github.io/bookmarklet-maker/`. The page title is "Bookmarklet Maker". A red arrow points from the "Reboot" bookmarklet icon in the browser's bookmark bar to the "Reboot" text in the "Bookmarklet:" field at the bottom of the page.

Bookmarklet Maker

Github Repository

Title:

Code

```
javascript:(function() {  
  const apiUrl = "https://ony39it269.execute-api.ap-northeast-2.amazonaws.com/prod/reboot";  
  const apiKey = "SDAEGT0W1guUI10uMok6kEyNFV14kZ5sE0wUzki";  
  
  fetch(apiUrl, {  
    method: "POST",  
    headers: {  
      "x-api-key": apiKey  
    }  
  })  
  .then(response => response.json())  
  .then(data => alert("Success: " + JSON.stringify(data)))  
  .catch(error => alert("Error: " + error));  
})
```

Bookmarklet:

Bookmarklet

The screenshot shows a web browser window with the following elements:

- Browser Tab:** "Bookmarklet Maker"
- Address Bar:** "caiorss.github.io/bookmarklet-maker/"
- Page Header:** "Reboot" with a refresh icon.
- Notification Box (Right):** "caiorss.github.io 내용: Success: {\"statusCode\":200,\"body\":\"\u2714Instance i-09dc8d08979526b68 rebooted successfully!\u2714\"}" with a "확인" (Check) button.
- Github Repository:** A section with a "Title" input field containing "Reboot".
- Code:** A text area containing the following JavaScript code:

```
javascript:(function() {  
  const apiUrl = "https://ony39it269.execute-api.ap-northeast-2.amazonaws.com/prod/reboot";  
  const apiKey = "SDAEGTcYlguUjDI0uMok6kEyNFVl4kZ5sEOwUzki";  
  
  fetch(apiUrl, {  
    method: "POST",  
    headers: {  
      "x-api-key": apiKey  
    }  
  })  
  .then(response => response.json())  
  .then(data => alert("Success: " + JSON.stringify(data)))  
  .catch(error => alert("Error: " + error));  
})
```
- Buttons:** "Generate Bookmarklet", "Run Code", and "Clear".
- Bookmarklet:** A blue button labeled "Reboot".

The screenshot shows a web browser window with two tabs: 'Bookmarklet Maker' and '널포유 개발용3 - 세명컴고 보안과 포털'. The address bar shows 'null3u.net'. The page content includes a navigation menu with items like '공지사항', '갤러리', '공모전 알람', '자유게시판', '자료실', '전공동아리', '수상현황', '보안뉴스', and '포털연혁'. The main header reads '널포유 개발용3 - 세명컴고 보안과 포털' and 'SMC SECU PORTAL'. Below this is a secondary navigation bar with the same menu items. The main content area features a news article titled 'Комета Казино Онлайн' (Comet Casino Online) by 'sca' on 'September 24, 2024'. The article text discusses online casino entertainment. To the right of the article is a calendar for 'January 2025' with dates 29 through 31. Below the calendar is a search bar with a 'Search' button. At the bottom right, there are several news snippets, including '2023 대입 BoB(Best Of Best) CTF mostbet mostbet AZ' and '대회 동아리'.