PROJECT

TOONIFY

클라우드를 이용한 웹호스팅
생성형 AI활용

sillacloud533

강시원(201937002) - https://github.com/telomere-K
배성윤(202090541) - https://github.com/Bae309
홍요한(202295035) - https://github.com/HongYohan

PROJECT

목차

PROJECT

프로젝트 소개

AWS를 이용하여 사진을 2D 애니메이션 사진으로 변환하는 웹페이지 만들기

PROJECT

프로젝트 목표

01

클라우드 응용 웹 사이트 제작 통한 학습

생성형 AI모델 이용을 위한 연결 구현

각 기능별 최적화

02

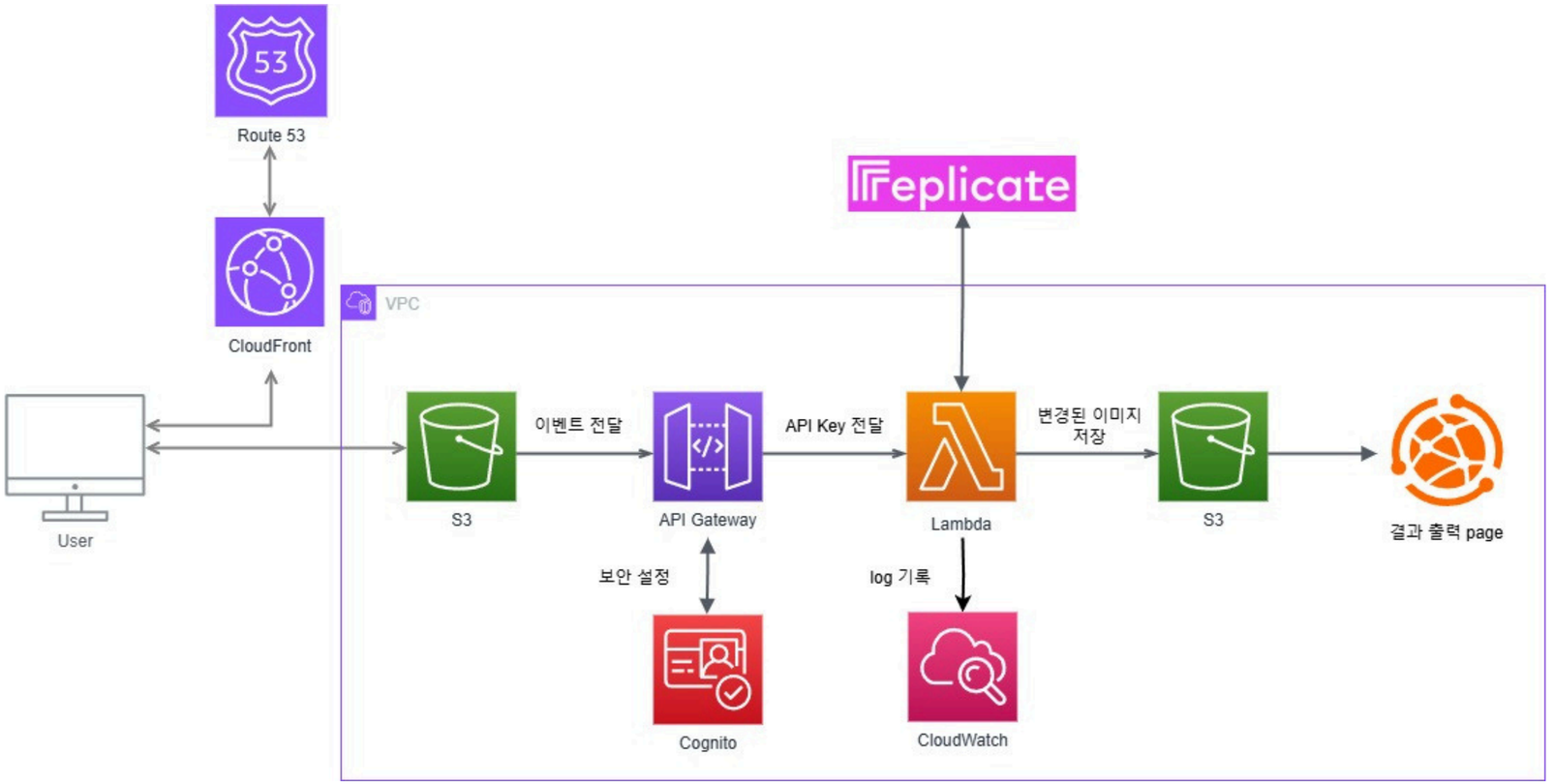
최소권한 정책 수행 확인

IAM User, Role, Policy

각 기능별 Policy, CORS 점검

PROJECT

프로젝트 서비스 개요도



PROJECT

프로젝트 진행과정 | 역할 분담

01

AI 모델 및 Lambda 담당(강시원)
IAM role 할당 및 Policy 설정

02

프론트엔드 개발 및 사용자
인터페이스 담당(배성윤)

03

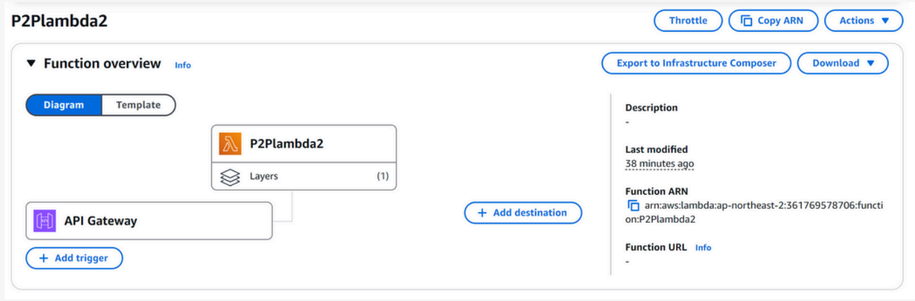
AWS 배포 및 최적화 담당(홍요한)

PROJECT

서버리스 환경 구성

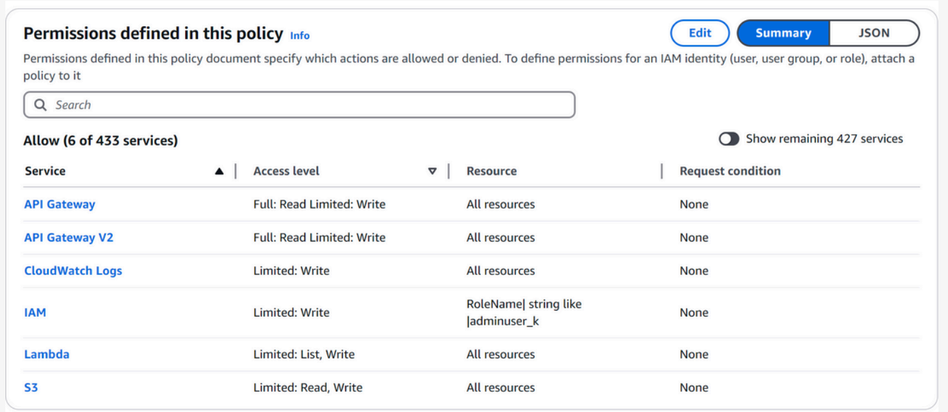
01

P2P 모델 연결 및 레이어구성



02

Policy 구성 및 IAM role 할당



PROJECT

PIX2PIX 모델 결정

https://github.com/junyanz/pytorch-CycleGAN-and-pix2pix

Pix2Pix 모델 구현 및 최적화
구동환경 확인 및 필요 성능 요구량, 결과물을 확인하기 위해 로컬에서 테스트를 진행

방안 1: EC2 및 SageMaker 활용

방안 2: Lambda 활용

방안 3: 외부 API 사용

결론 :
단기간 내의 실시간 서비스가 중요한 클라우드 환경에서 필요로 되는 것은 빠른 처리, 변수 방지를 위해 만들어진 외부 API사용

PROJECT

LAMBDA 구현

S3 - API Gateway - Lambda - 외부 API 지정

```
def lambda_handler(event, context):
    print(f"Received event: {event}")  # Debugging: 로그로 전체 이벤트 출력

    # S3 및 API 설정
    s3 = boto3.client('s3')
    input_bucket = "syinputimage"  # 입력 버킷 이름
    output_bucket = "syindexbucket"  # 출력 버킷 이름
    api_key = "r8_SwODbWV4kkqW4KKw13g4ugVh7zbKpOX1V3m6z"  # Replicate API Key
    model_url = "https://api.replicate.com/v1/predictions"

    # 요청에서 body 가져오기
    body = event.get('body', None)
    if not body:
        return {
            'statusCode': 400,
            'body': json.dumps({'message': 'Request body is required.'})
        }
```

Key 값을 이용한 S3 to API 연결

```
# 요청에서 Key 및 Prompt 추출
key = body.get('key')  # 예: "uploads/example-image.jpg"
prompt = body.get('prompt', "Anime Style, FHD photo")

if not key:
    return {
        'statusCode': 400,
        'body': json.dumps({'message': '"key" is a required field.'})
    }

# Key 존재 여부 확인
try:
    s3.head_object(Bucket=input_bucket, Key=key)
except s3.exceptions.ClientError:
    return {
        'statusCode': 404,
        'body': json.dumps({'message': f"The object '{key}' does not exist in bucket '{input
    }
```

PROJECT

LAMBDA 구현

모델 요구사항 충족 및 호출

```
# 이미지 데이터 Base64 인코딩
image_base64 = base64.b64encode(image_data).decode('utf-8')

# Replicate API 요청 데이터
input_data = {
    "version": "a3d3e0bdeea4925a873179e55701e1091e4b4d7ddeee9a205b932d9de1d9f181",
    "input": {
        "image": f"{image_base64}",
        "prompt": prompt
    }
}

# Replicate API 호출
try:
    headers = {
        "Authorization": f"Bearer {api_key}",
        "Content-Type": "application/json"
    }
    response = requests.post(model_url, headers=headers, json=input_data, timeout=30)
    response.raise_for_status()  # API 요청 상태 코드 확인
except requests.exceptions.RequestException as e:
    print(f"Error calling Replicate API: {e}")
    return {
        'statusCode': 500,
        'body': json.dumps({'message': 'Error calling Replicate API.', 'error': str(e)})
    }
```

S3 반환 및 출력을 위한 위치 지정 저장

```
# 출력 이미지를 S3에 저장
output_urls = []
try:
    for index, image_url in enumerate(output_data):
        img_response = requests.get(image_url, timeout=10)
        if img_response.status_code == 200:
            output_key = f"results/{key.split('/')[-1].split('.')[0]}_output_{index}.png"
            s3.put_object(
                Bucket=output_bucket,
                Key=output_key,
                Body=img_response.content,
                ContentType="image/png"
            )
            output_urls.append(f"https://{output_bucket}.s3.amazonaws.com/{output_key}")
        else:
            print(f"Failed to download image from {image_url}")
except Exception as e:
    print(f"Error saving images to S3: {e}")
    return {
        'statusCode': 500,
        'body': json.dumps({'message': 'Failed to save processed images to S3.', 'error': str(e)
    }
```

PROJECT

CW LOGS 확인

Logs 관찰 결과 정상 작동하는 모습 확인 가능

```
2024-12-06T06:39:12.935Z    INIT_START Runtime Version: python:3.13.v13 Runtime Version ARN: arn:aws:lambda:ap-northeast-2::runtime:b881cbc9a10a8…

INIT_START Runtime Version: python:3.13.v13    Runtime Version ARN: arn:aws:lambda:ap-northeast-
2::runtime:b881cbc9a10a8bcb3def9d9e9fe38f922bb36510a1d92d4ce85cf2a899eeabd8
```

```
2024-12-06T06:39:13.466Z    START RequestId: 5fbdf45f-eefa-42ca-ac3f-baa559a16507 Version: $LATEST

START RequestId: 5fbdf45f-eefa-42ca-ac3f-baa559a16507 Version: $LATEST
```

```
2024-12-06T06:39:13.466Z    === Lambda Function Invoked ===

=== Lambda Function Invoked ===
```

```
2024-12-06T06:39:18.413Z    END RequestId: 5fbdf45f-eefa-42ca-ac3f-baa559a16507

END RequestId: 5fbdf45f-eefa-42ca-ac3f-baa559a16507
```

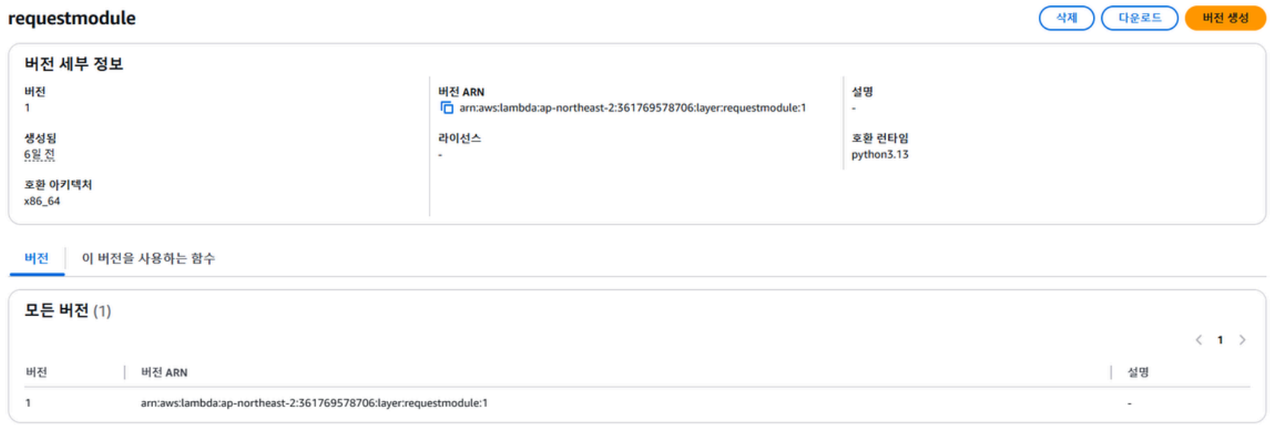
```
2024-12-06T06:39:18.414Z    REPORT RequestId: 5fbdf45f-eefa-42ca-ac3f-baa559a16507 Duration: 4946.86 ms Billed Duration: 4947 ms Memory Size: 128…

REPORT RequestId: 5fbdf45f-eefa-42ca-ac3f-baa559a16507  Duration: 4946.86 ms    Billed Duration: 4947 ms    Memory Size: 128 MB    Max
Memory Used: 94 MB    Init Duration: 528.50 ms
```

PROJECT

API 연결 위한 람다 설정

해당 레이어 등록 방식을 통해 람다 간 공통으로 사용되는 모듈을 쉽게 관리하고 추가 가능

각 레이어별 저장 공간을 통해 단일 업로드 대비 모듈 크기 제한을 우회 가능

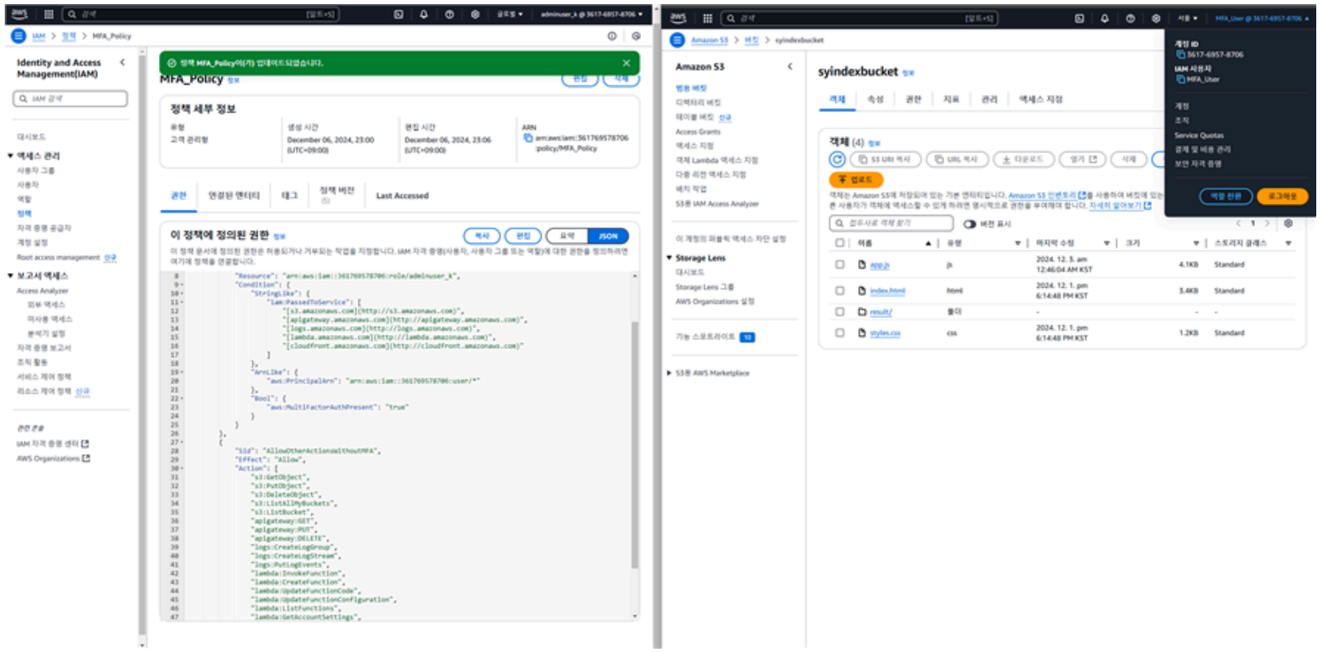


PROJECT

IAM ROLE 설정

추후 프로젝트 확장을 위해 Root 유저와 admin 권한을 가진 인원 외 초보 사용자가 추가 투입될 때 기본 작업 외에도 추가적 권한이 필요하다면 MFA 를 통해 제한적 허용을 가능하게 합니다.

필요 권한을 확인하고 추가한 뒤, 계정전환을 통해 권한을 갱신하여 결과를 추적할 수 있다.



PROJECT

웹 애플리케이션 개발

01

사용자 이미지 업로드 기능 구현

02

변환 결과 이미지 표시 UI 개발

PROJECT

Toonify

Upload your photo:

파일 선택 example.jpg

Convert to 2D Art

PROJECT

Upload your photo:

파일 선택 example.jpg

Convert to 2D Art

Converting... Please wait

남은 시간: 26초

PROJECT

Upload your photo:

파일 선택 example.jpg

Convert to 2D Art

Converted Image

PROJECT

반응형 디자인

01

모바일 및 데스크톱 환경에서 사용 가능

02

Tailwind CSS를 이용한 구현

PROJECT

반응형 디자인

```
<BODY CLASS="BG-GRAY-100 MIN-H-SCREEN FLEX FLEX-COL ITEMS-CENTER">

<NAV CLASS="BG-BLUE-500 TEXT-WHITE P-4 W-FULL">
    <DIV CLASS="CONTAINER MX-AUTO">
        <H1 CLASS="TEXT-2XL FONT-BOLD">PHOTO TO 2D ART</H1>
    </DIV>
</NAV>

<FORM ID="UPLOADFORM" CLASS="BG-WHITE P-6 SHADOW-MD ROUNDED-MD FLEX
FLEX-COL ITEMS-CENTER">
```

PROJECT

반응형 디자인

```
<IMG SRC="${E.TARGET.RESULT}" ALT="PREVIEW" CLASS="MAX-W-XS MX-AUTO SHADOW-MD ROUNDED-MD" />

<IMG ID="CONVERTEDIMAGE" SRC="" ALT="CONVERTED" CLASS="MX-AUTO SHADOW-MD ROUNDED-MD" />
```

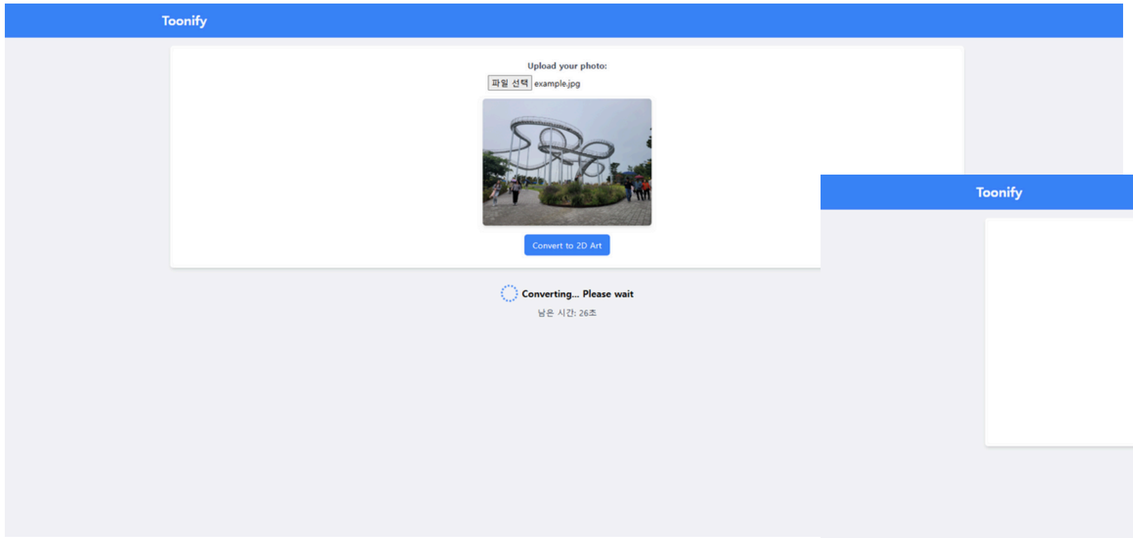
PROJECT

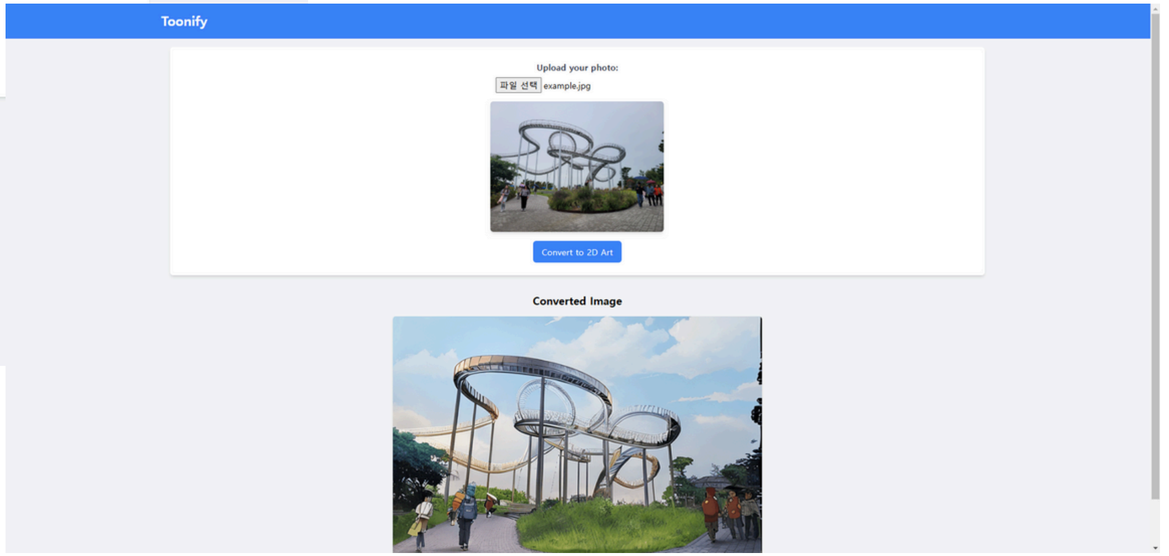
반응형 디자인

```
<BUTTON CLASS="BG-BLUE-500 TEXT-WHITE PX-4 PY-2 ROUNDED-MD HOVER:BG-BLUE-600 SM:PX-6 SM:PY-3">
    CONVERT TO 2D ANIMATION
</BUTTON>

<DIV CLASS="CONTAINER MX-AUTO P-4 SM:P-8 LG:W-2/3">

<H2 CLASS="TEXT-XL FONT-BOLD MB-4 SM:TEXT-2XL">CONVERTED IMAGE</H2>
```

PROJECT

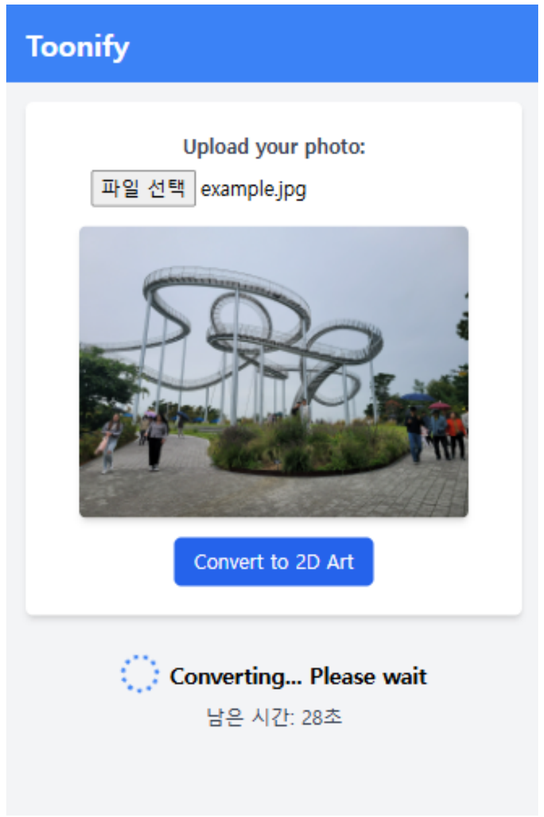
반응형 디자인

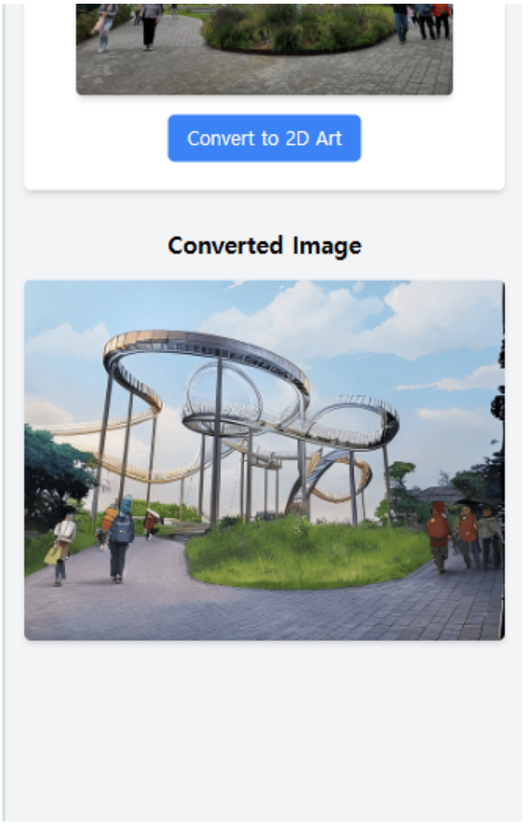




PROJECT

반응형 디자인





PROJECT

AWS 리소스 최적화

01

성능 최적화

1. 메모리 및 CPU 할당 조정
(AWS Lambda Power Tuning 도구 활용)
2. 불필요한 초기화 코드 제거
3. AWS SDK를 최신 버전으로 유지(성능 개선)
4. 외부 API 호출을 병렬 처리하도록 설계
5. 함수 패키지 최적화

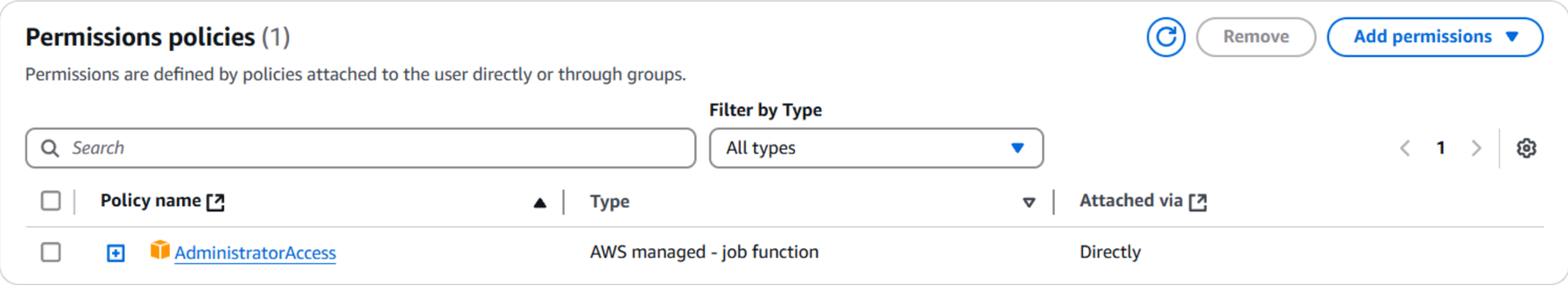
02

1. Lambda 성능 최적화
2. S3 성능 최적화
3. API Gateway 성능 최적화

PROJECT

lambda 성능 최적화

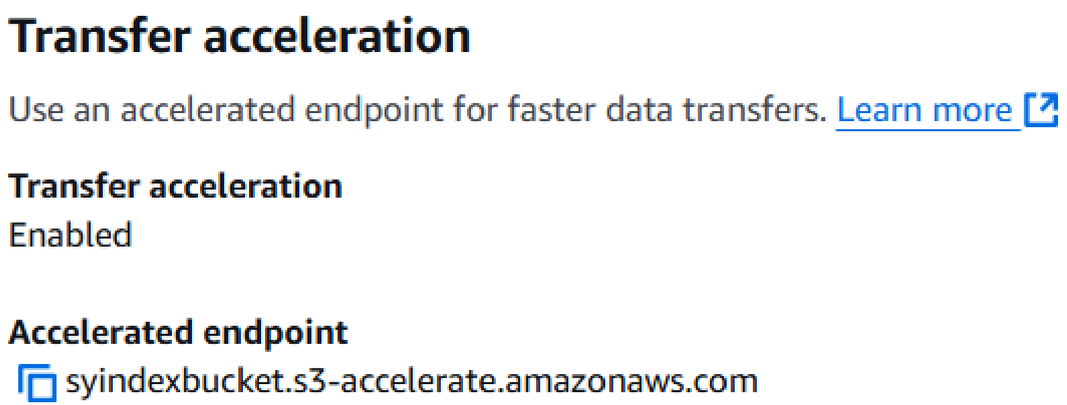
- Lambda 함수에 필요한 권한만 부여하도록 IAM 정책을 최소화합니다.
- AWS Managed Policies를 사용하되 필요하다면 Custom Policy 작성.
- Lambda Execution Role에 불필요한 서비스 접근 권한이 포함되어 있지 않은지 확인.



PROJECT

S3 성능 최적화

- 데이터를 빠르게 업로드/다운로드하려면 Transfer Acceleration 활성화.
- S3 - Buckets - Properties -Transfer acceleration - Edit - Enabled로 설정
- S3 객체를 CloudFront와 연동하여 캐싱을 활성화하면 웹사이트의 성능을 향상시킬 수 있다.



PROJECT

API 성능 최적화

- API Gateway의 Usage Plans를 설정하여 요청 제한(Throttle Rate)을 적용.

The CORS configuration, written in JSON, defines a way for client web applications that are loaded in one domain to interact with resources in a different domain. Learn more

Copy

```
[
  {
    "AllowedHeaders": [
      "*"
    ],
    "AllowedMethods": [
      "GET",
      "PUT",
      "POST",
      "DELETE",
      "HEAD"
    ],
    "AllowedOrigins": [
      "*"
    ],
    "ExposeHeaders": [
      "ETag",
      "x-amz-request-id"
    ],
    "MaxAgeSeconds": 3000
```

Additional settings

Cache settings Info

You can enable API caching to cache your endpoint's responses. With caching, you can reduce the number of calls made to your endpoint and also improve the latency of requests to your API. Caching is charged by the hour based on cache size, see API Gateway pricing for details.

Provision API cache
Provision API caching capabilities for your stage. Caching is not active until you enable the method-level cache.

Default method-level caching
Activate method-level caching for all GET methods in this stage.

Cache capacity

0.5GB

Encrypt cache data

Cache time-to-live (TTL)

seconds

3600

Must be between 0-3600 seconds.

PROJECT

CLOUD FRONT 배포

01

1. S3 버킷에서 정적 웹 호스팅 콘텐츠 활성화
2. Cloud 배포 생성
3. Route 53 생성(Hosted zone)
4. AWS Certificate Manager에서 허가

02

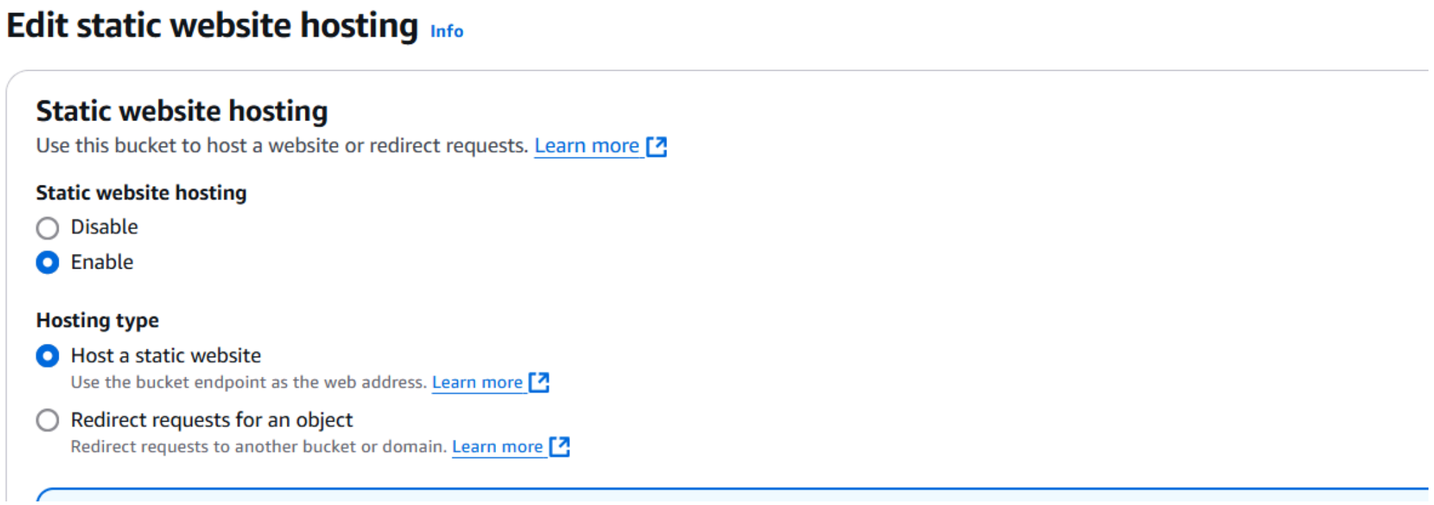
최적화 및 검토

1. CloudFront에서 자주 변경되지 않는 정적 콘텐츠는 더 긴 TTL 설정(예: 이미지, CSS).
2. Standard Logging을 활성화하여 액세스 로그를 S3 버킷에 저장.

PROJECT

S3 버킷에서 정적 웹 호스팅 활성화

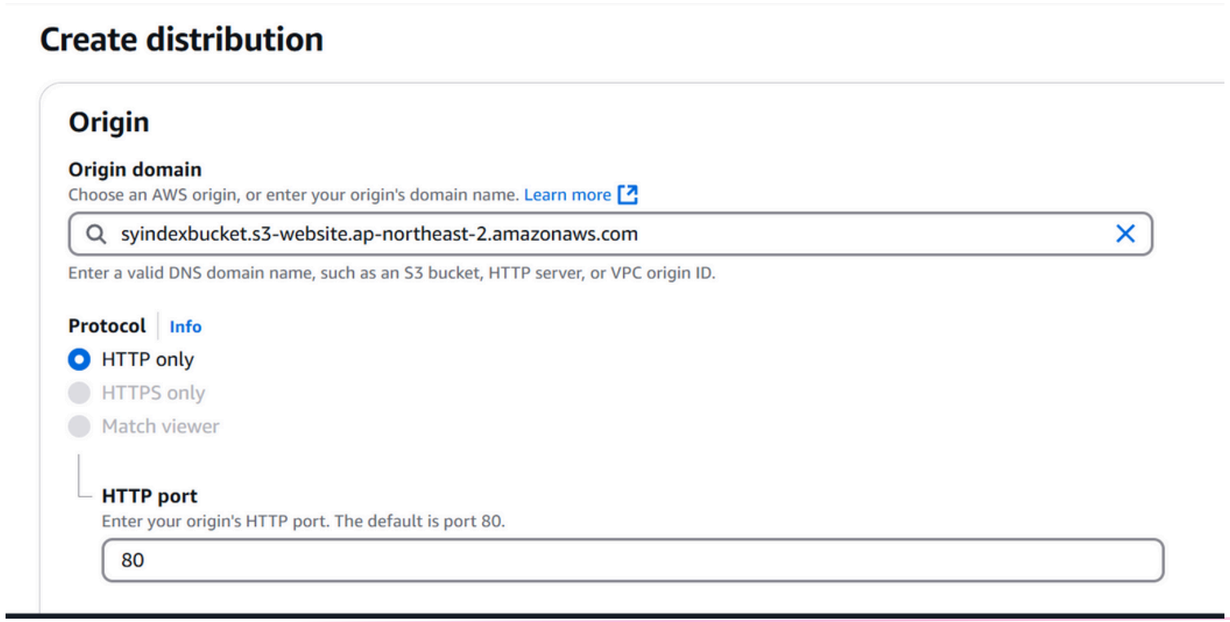
- 정적 웹 호스팅은 HTML, CSS, JavaScript, 이미지 등 정적 파일을 제공하는 데 최적화되어 있습니다.
- 다른 서비스(예: EC2, Lambda 등)보다 저렴한 가격으로 파일을 호스팅할 수 있습니다.
- 높은 확장성을 제공하는 방식으로 빠르고 간편하게 정적 웹사이트를 배포할 수 있습니다.
- S3 Bucket - Properties - Static website hosting - Edit



PROJECT

CloudFront 생성

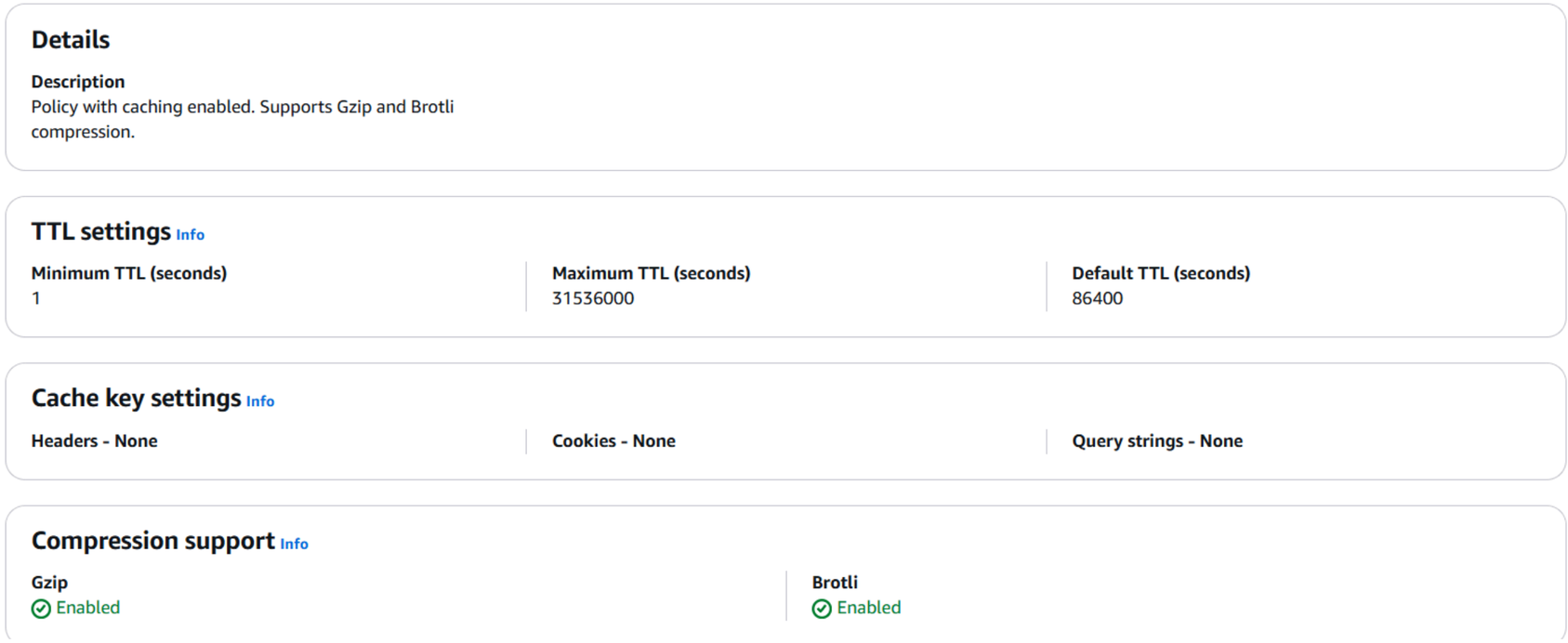
- Origin Domain: S3 버킷의 정적 웹 호스팅 엔드포인트 입력
- Restrict Bucket Access: No 선택 (정적 웹 호스팅에 퍼블릭 접근이 허용된 경우).
- Viewer Protocol Policy: Redirect HTTP to HTTPS 선택
- Allowed HTTP Methods: GET, HEAD 선택(정적 콘텐츠에 적합).



PROJECT

Cache Behavior Settings 생성

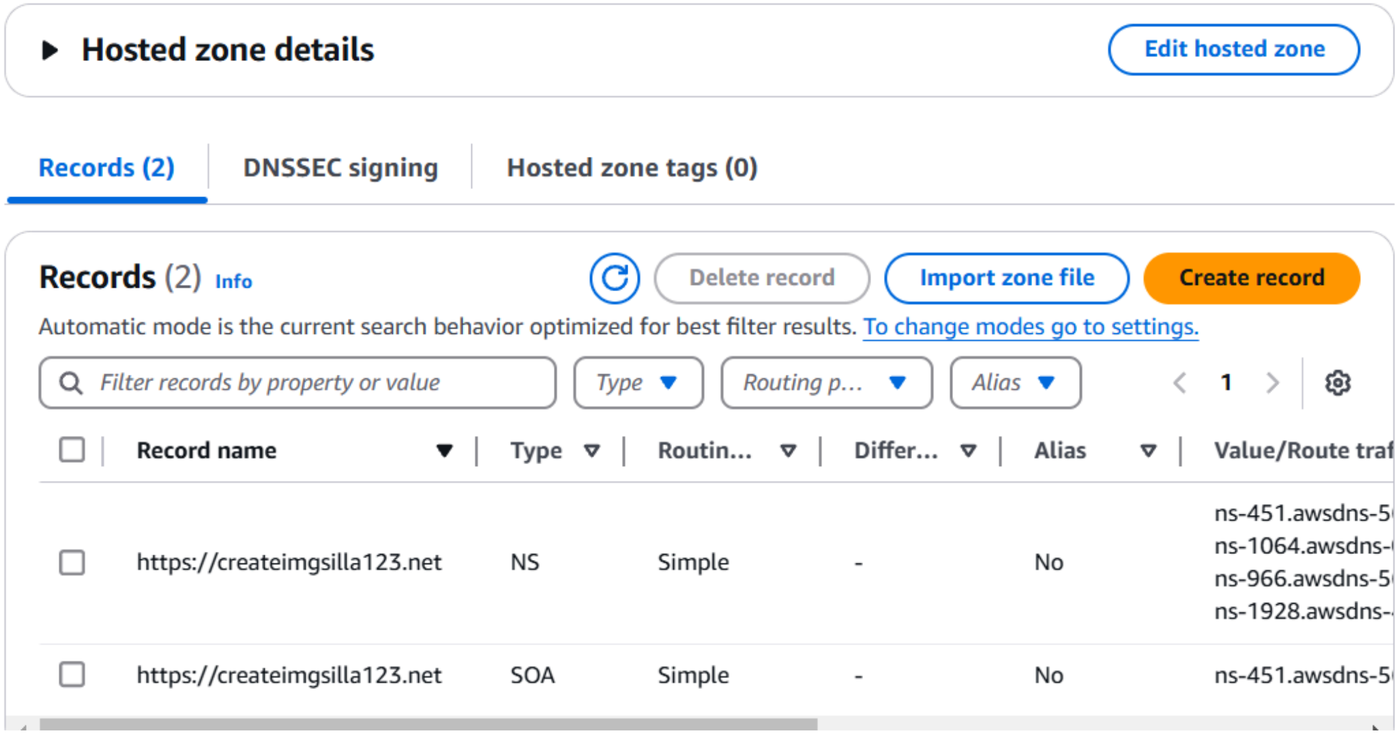
- Default TTL: 캐싱 기간 설정 (예: 3600초 or Defalut(*)).
- Compress Objects Automatically: 활성화(Gzip 압축을 통해 대역폭 최적화).
- Query String Forwarding: 필요 시 활성화(예: 동적 콘텐츠 처리 시).



PROJECT

Rount 53에서 Hosted Zone 생성

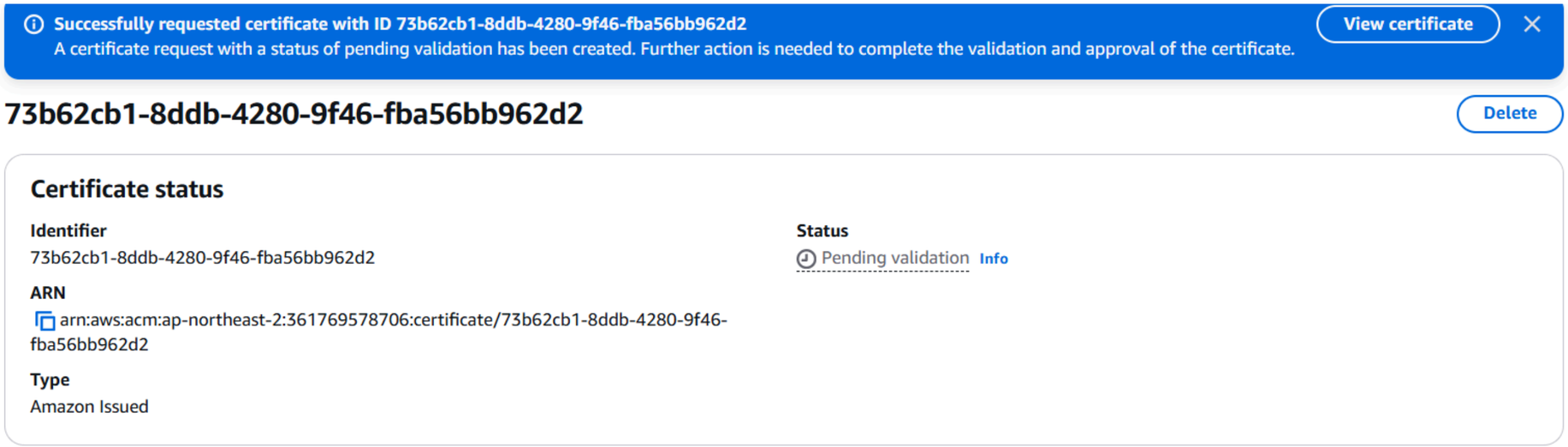
- 원하는 사용자 도메인을 생성하면 된다.
- 대신 인증을 받아놔야 한다. 그래서 AWS Certificate Manager로 이동



PROJECT

AWS Certificate Manager에서 certification 생성

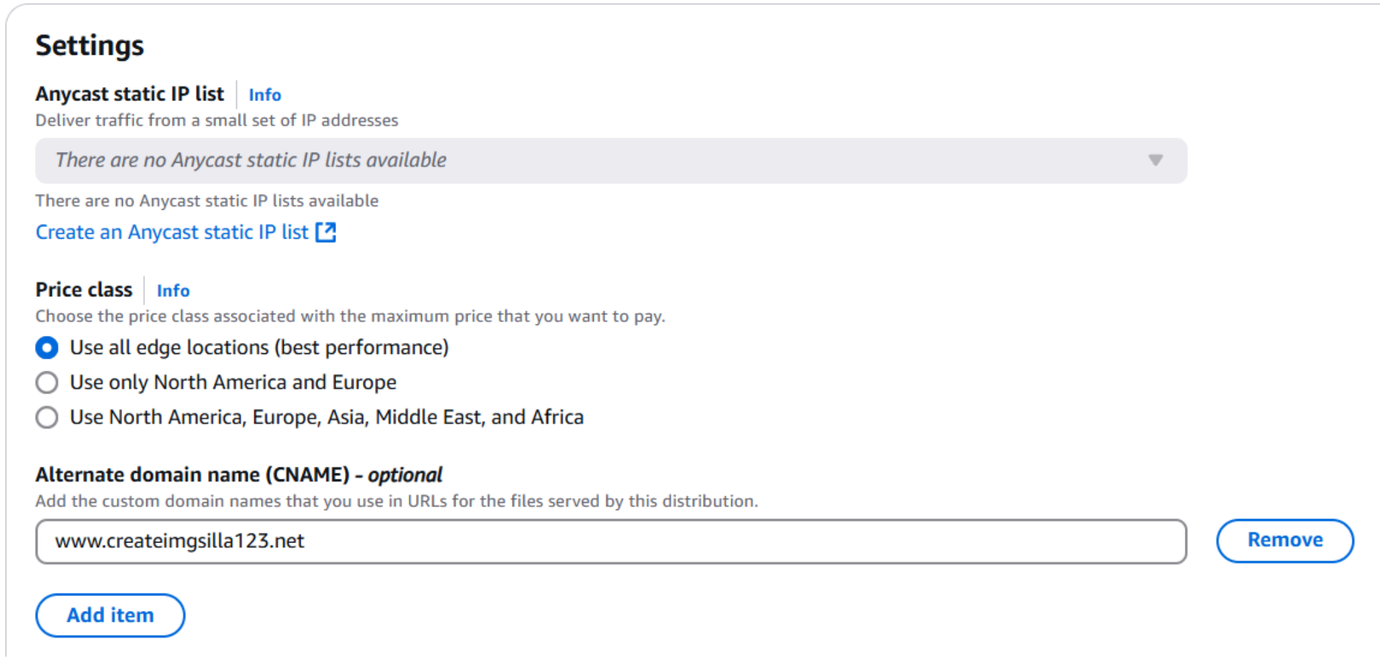
- Hosted Zone에서 생성한 도메인 명과 같이 해서 새로 생성한다.
- 인증서를 생성할 AWS 리전을 미국 동부 (N. Virginia)로 선택해야 합니다.(미국 동부만 허가됨)



PROJECT

CloudFront에서 CNAME을 추가

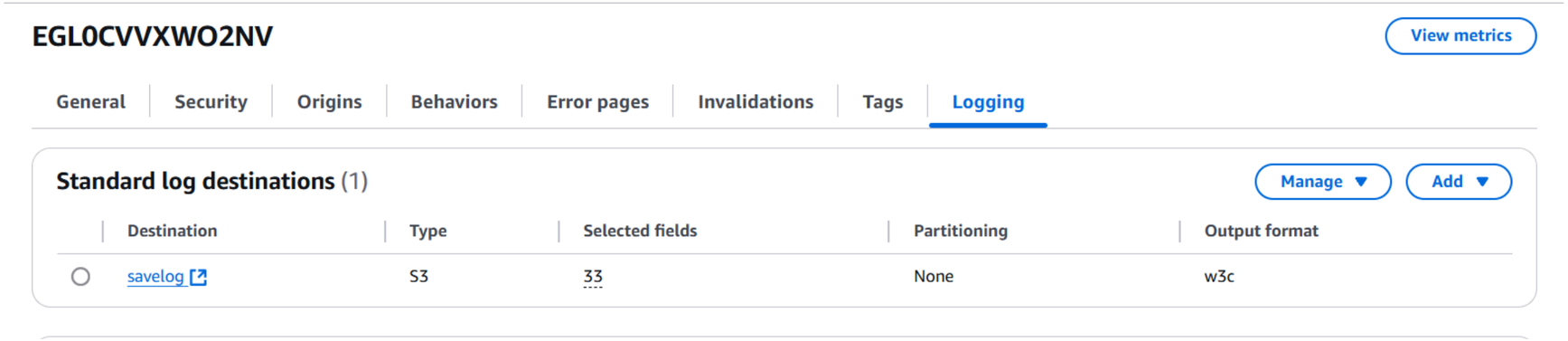
- CloudFront - General - Setting에서 CNAME에 도메인 명 작성
- Route 53에서 Alias를 생성하고 해당 도메인을 확인하면 된다. (그러나 Certification을 못받았다..)



PROJECT

CloudFront에서 Standard Logging 설정

- CloudFront의 로그 기록을 알기 위해서 특정 S3 버킷 생성
- CloudFront - General - Standard log destinations에 특정 S3 버킷 추가
- 그렇게 해서 최적화를 수행한다.



PROJECT

보안 설정

01

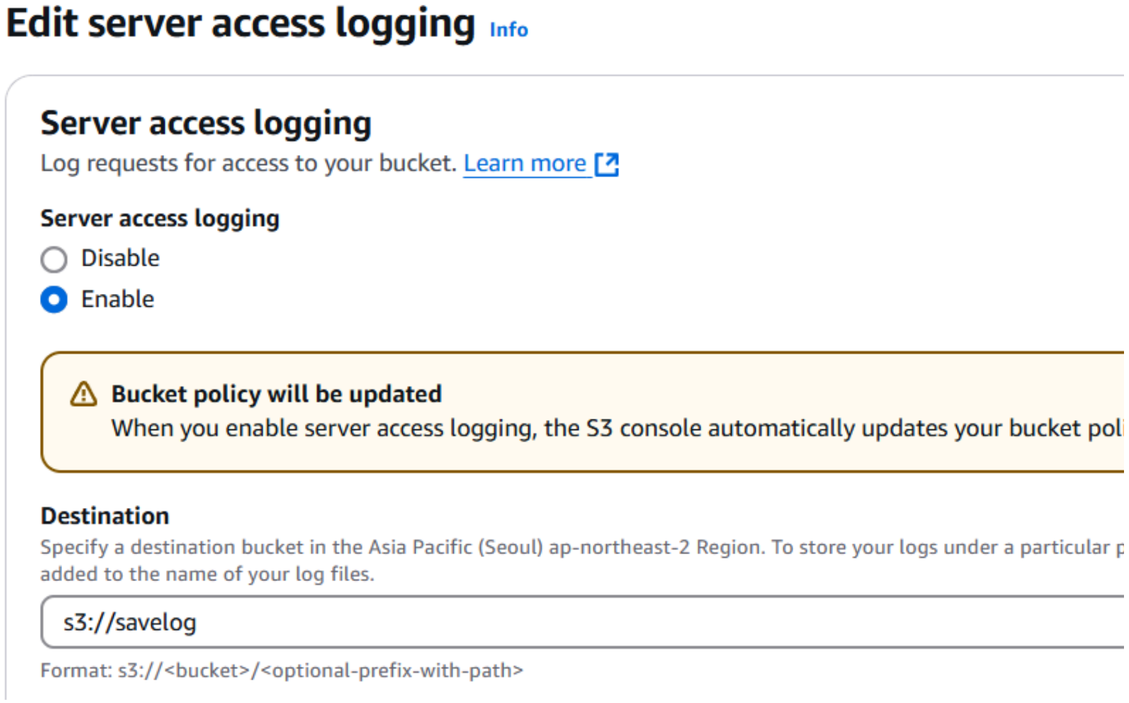
1. S3 버킷 정책 검토
2. API Gateway 보안 구성
3. Lambda 보안그룹 구성
4. 추가적인 보안 조치

02

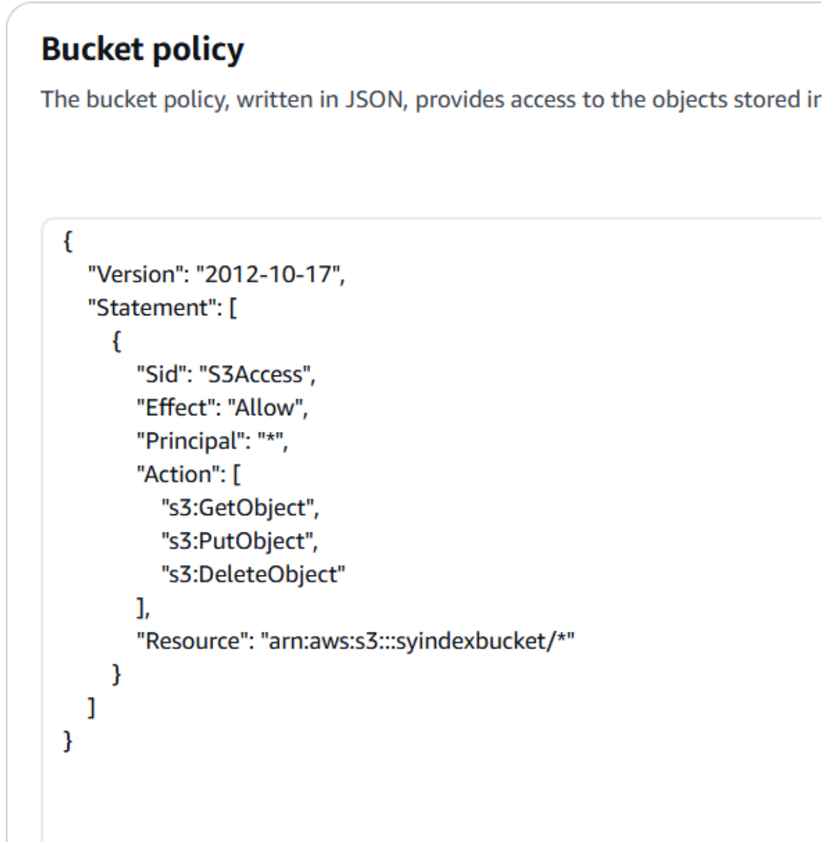
PROJECT

S3 버킷 정책 검토

- 액세스 로그 활성화
- Propertise - Server access logging - Edit



- 버킷 정책 검토 및 수정



PROJECT

- API Gateway 보안 구성 (JWT 또는 OAuth 2.0 인증)

My web app - 0ct-4v

Names are limited to 128 characters or fewer. Names may only contain alphanumeric characters, spaces, and the following special characters: + = , . @ -

Configure options

You must make a few initial choices about the user pool that supports your application. To change these settings later, you must create a new user pool.

Options for sign-in identifiers | Info

Choose sign-in attributes. Usernames can be an email address, phone number, or a user-selected username. When you select only email and phone, users must select either email or phone as their username type. When username is an option, users can sign in with any options you select if they have provided a value for that option.

- [x] Email
- [ ] Phone number
- [ ] Username

Want to set up social, SAML, or OIDC sign-in?

Required attributes for sign-up | Info

Choose any attributes that you want to require users to provide. With username alone, you must set email address or phone number as a required attribute.

Select Attributes

email
User's preferred email address.

⚠ Options for sign-in identifiers and required attributes can't be changed after the app has been created.

Add a return URL - *optional*

Choose a return URL. Cognito redirects to this URL after successful sign-in with the managed login pages on your user pool domain. Your application can then process th

Return URL | Info

Allowed callback URLs | Info

Enter at least one callback URL to redirect the user back to after authentication. This is typically the URL for the app receiving the authorization code issued by Cognito. schemes.

URL

https://createimgsilla123.net

Length of callback URL must be between 1 and 1024 characters. Valid characters are letters, marks, numbers, symbols, and punctuations. Amazon Cognito requires HTTPS over HTTP except for http://localhost for testing purposes only. App callback URLs such as myapp://example are also supported. Must not contain a fragment.

Add another URL

You can add 99 more URLs

Allowed sign-out URLs - *optional* | Info

Enter at least one sign-out URL. The sign-out URL is a redirect page sent by Cognito when your application signs users out. This is needed only if you want Cognito to di callback URL.

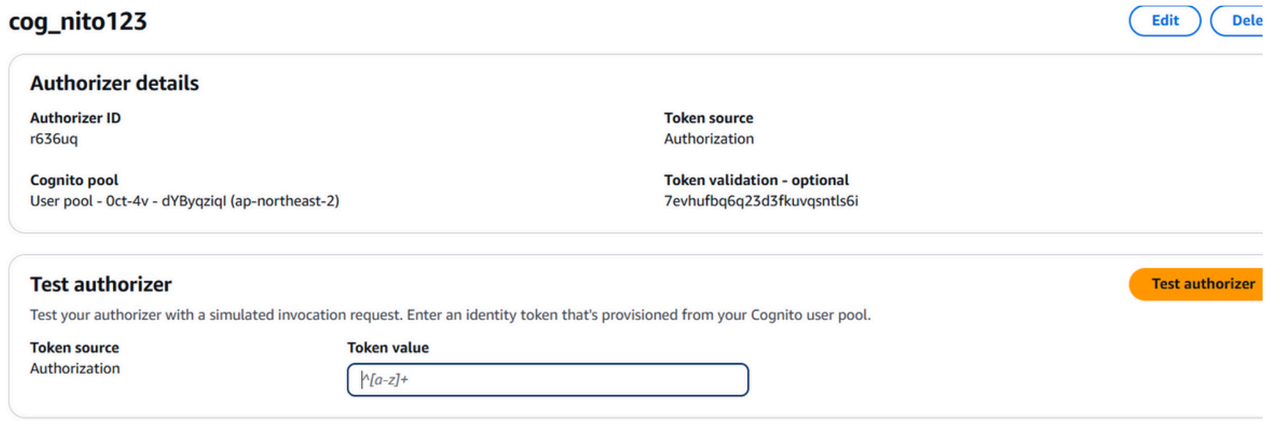
URL

https://createimgsilla123.net

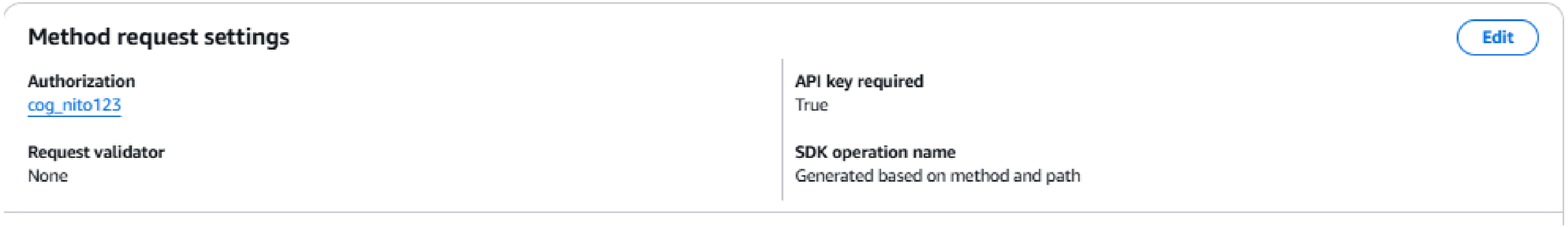
Length of sign-out URL must be between 1 and 1024 characters. Valid characters are letters, marks, numbers, symbols, and punctuations. Amazon Cognito requires HTTPS over HTTP except for http://localhost for testing purposes only. App sign-out URLs such as myapp://example are also supported. Must not contain a fragment.

PROJECT

- JWT 또는 OAuth 2.0 인증



- API Gateway에서 Cognito 설정 및 API Key Required를 TRUE로 설정



Lambda 보안그룹 구성

PROJECT

- VPC 설정

Subnets (1/8) Info

Find resources by attribute or tag

Name	Subnet ID	State	VPC	Block Public...	IPv4 CIDR	IPv6 CIDR	IPv6 CIDR association ID
RDS-Pvt-subnet-3	subnet-006edefc4c154cdb8	Available	vpc-01dedf840068b7061	Off	172.31.65.0/25	–	–
–	subnet-0da78459498f10766	Available	vpc-01dedf840068b7061	Off	172.31.16.0/20	–	–
RDS-Pvt-subnet-4	subnet-04d7f810683374ebf	Available	vpc-01dedf840068b7061	Off	172.31.65.128/25	–	–
RDS-Pvt-subnet-2	subnet-03b26ac2a8f9a40a3	Available	vpc-01dedf840068b7061	Off	172.31.64.128/25	–	–
RDS-Pvt-subnet-1	subnet-0539fc900431331b6	Available	vpc-01dedf840068b7061	Off	172.31.64.0/25	–	–

- VPC 설정(Route tables 생성)

Route tables (1/2) Info

Find resources by attribute or tag

Name	Route table ID	Explicit subnet associ...	Edge associations	Main	VPC	Owner ID
RDS-Pvt-rt	rtb-01370b6ab6c75d596	4 subnets	–	No	vpc-01dedf840068b7061	361769578706
–	rtb-02ee2277c5f0389c1	–	–	Yes	vpc-01dedf840068b7061	361769578706

- VPC 설정(NAT gateways 설정)

VPC > NAT gateways > nat-004d3e033ad5f9d2a

nat-004d3e033ad5f9d2a / RDS-Pvt-subnet-4 Actions

Details

NAT gateway ID
nat-004d3e033ad5f9d2a

NAT gateway ARN
arn:aws:ec2:ap-northeast-2:361769578706:natgateway/nat-004d3e033ad5f9d2a

VPC
vpc-01dedf840068b7061

Connectivity type
Public

Primary public IPv4 address
–

Subnet
subnet-04d7f810683374ebf / RDS-Pvt-subnet-4

State
Pending

Primary private IPv4 address
–

Created
Wednesday, December 4, 2024 at 19:47:25 GMT+9

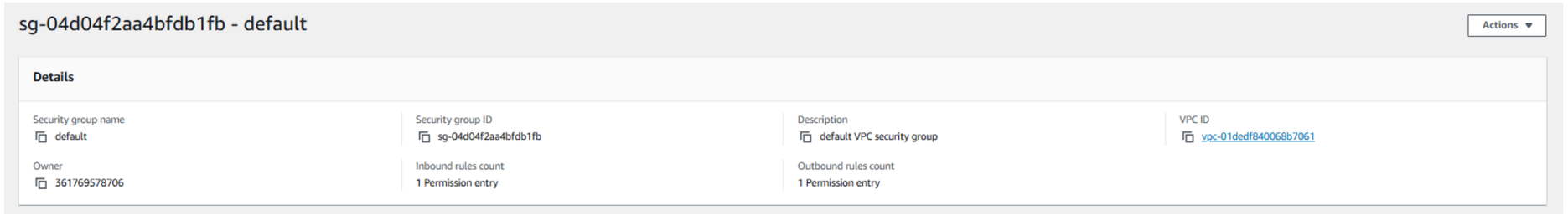
State message Info
–

Primary network interface ID
–

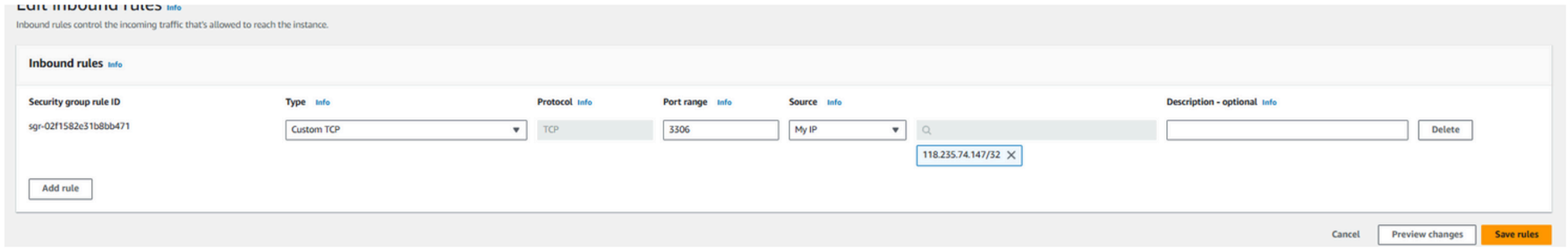
Deleted
–

PROJECT

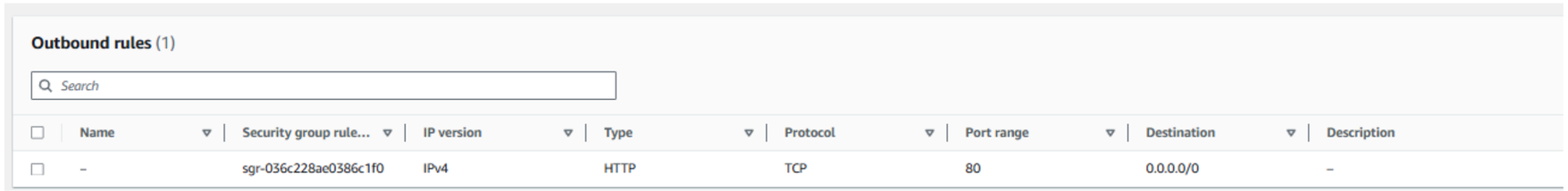
- VPC 설정(보안그룹 생성)



- VPC 설정(Inbound rules 설정)



- VPC 설정(Outbound rules 설정)

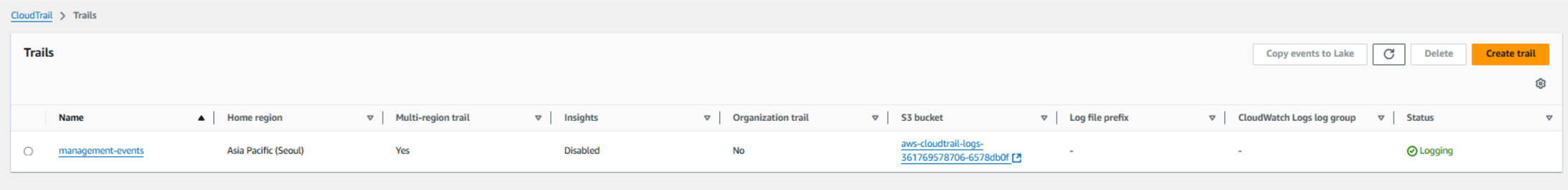


PROJECT

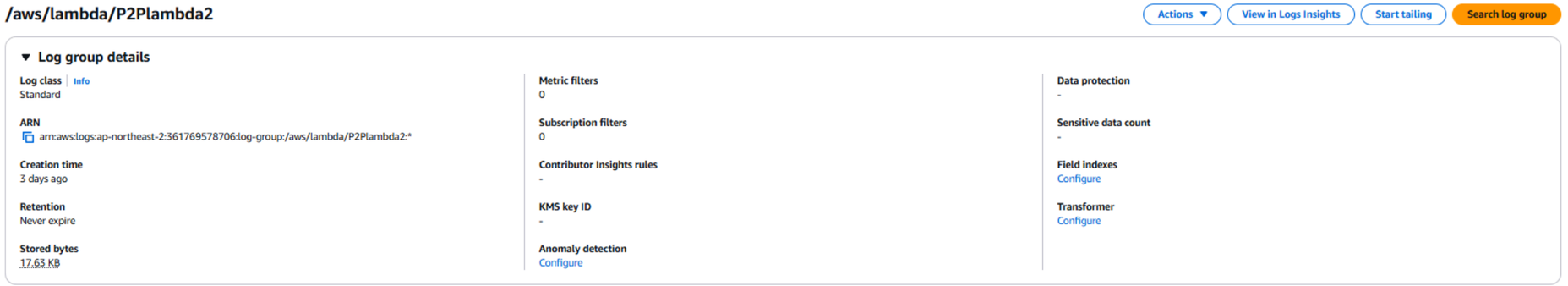
추가적인 보안 조치

- CloudTrail(API 호출과 이벤트 기록을 저장하여, 보안 감사, 규정 준수, 문제 해결)
- CloudWatch Logs(실시간 상태 모니터링)
- S3, API Gateway, Lambda 암호화

- CloudTrail 설정



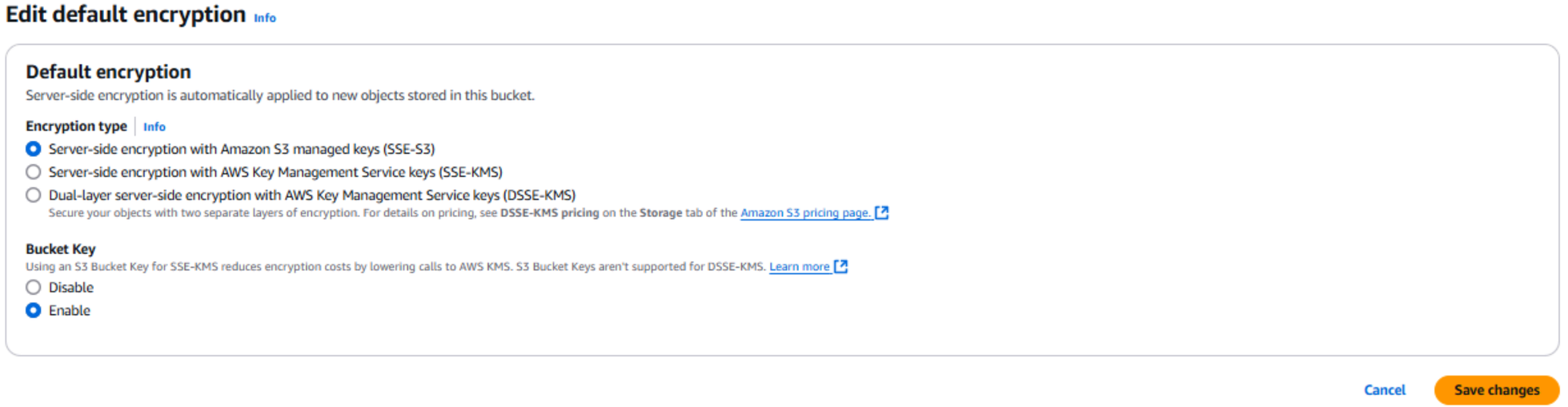
- CloudWatch Logs 설정



PROJECT

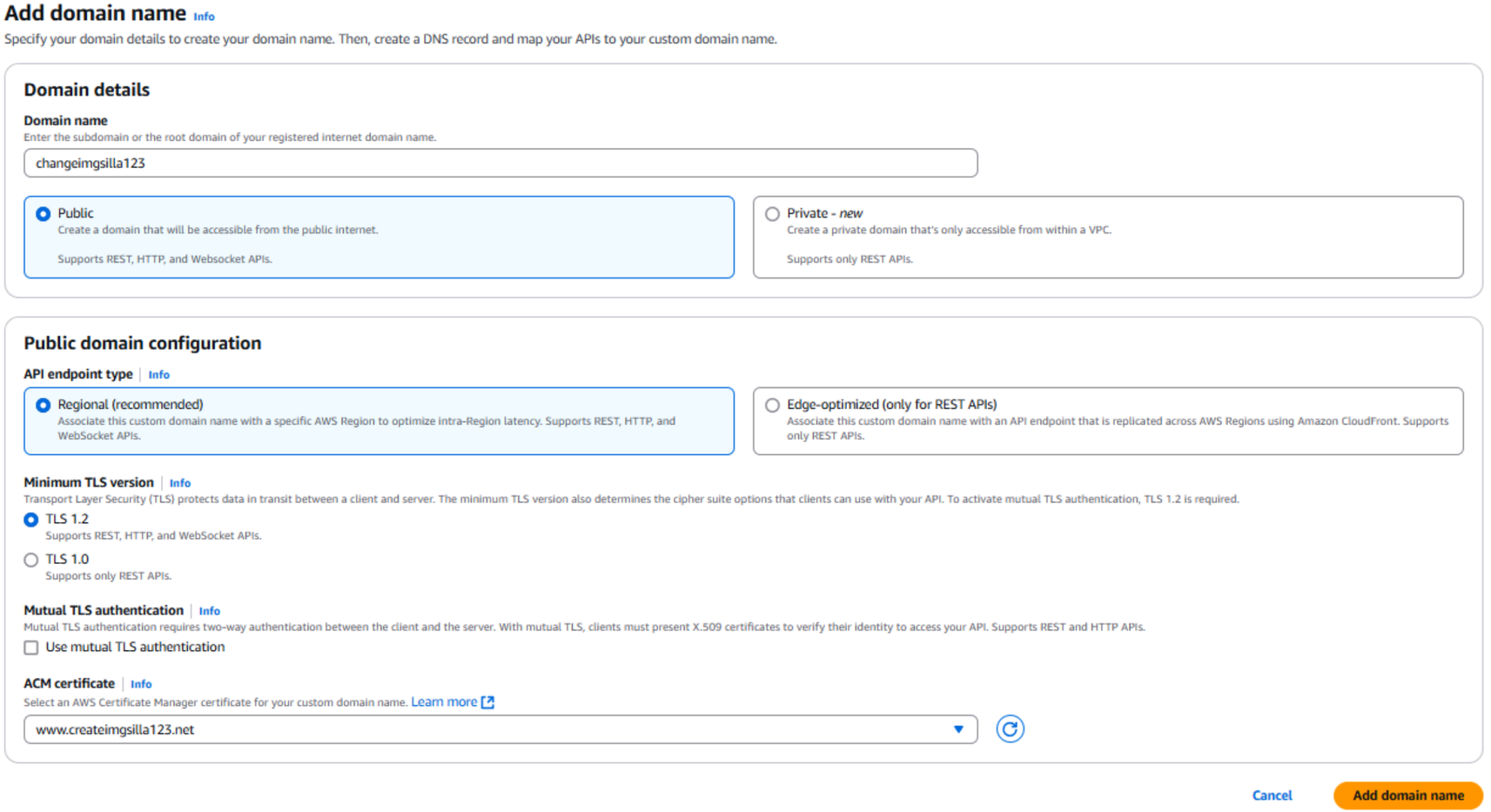
S3, API Gateway, Lambda 암호화

- S3 암호화



PROJECT

- API Gateway 암호화(HTTP 활성화)



PROJECT

- API Gateway 암호화(HTTP 활성화)

Customer managed keys (1)

Filter keys by properties or tags

Aliases	Key ID	Status	Key type
syntexbucket	d222544e-2425-4997-97d8-20f7959c2431	Enabled	Symmetric

- Lambda 환경 변수 암호화

Edit environment variables

Environment variables

You can define environment variables as key-value pairs that are accessible from your function code. These are useful to store configuration settings without the need to change function code. Learn more

There are no environment variables on this function.

Add environment variable

Encryption configuration

Encryption in transit Info

Enable helpers for encryption in transit

AWS KMS key to encrypt at rest

Choose an AWS KMS key to encrypt the environment variables at rest, or simply let Lambda manage the encryption.

(default) aws/lambda

Use a customer master key

Customer master key

arn:aws:kms:ap-northeast-2:361769578706:key/d222544e-2425-4997-97d8-20f7959c2431

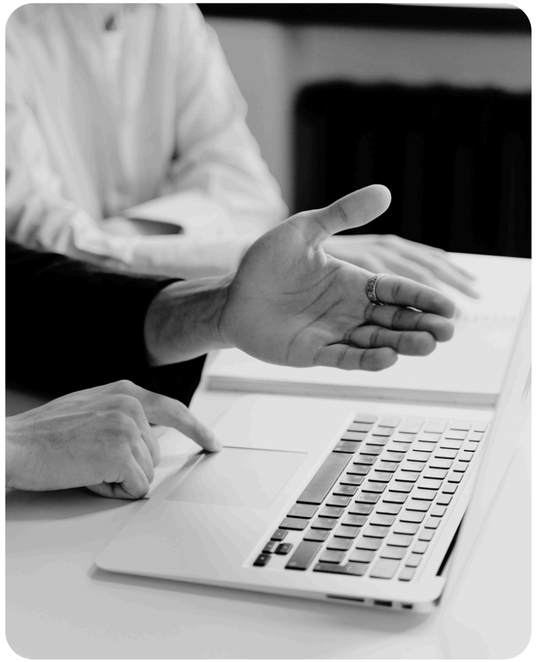
Cancel Save

PROJECT

결론

각 단계별 개선의 방향성을 파악 할 수 있었음

추후 발전 시 SageMaker의 Notebook 활용,
EC2 – S3 – SageMaker로 연동하여 모델 학습을
통한 서비스 차별화 모색



Q&A

궁금한 점 질문해 주세요.

THANK YOU

배성윤, 강시원, 홍요한