

Team *Epícer*

ECS 기반 고가용성과 테라폼을 이용한 레시피 공유 플랫폼 구축

김재원 허은준

목 차

1. 팀원소개
2. 시나리오 및 고객 요구사항
3. 아키텍처
4. 프로젝트 구현

팀원소개

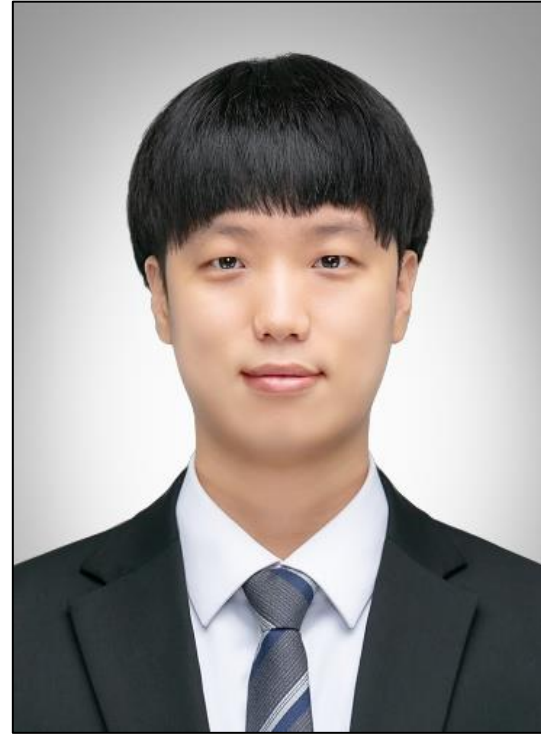


팀원소개



김재원

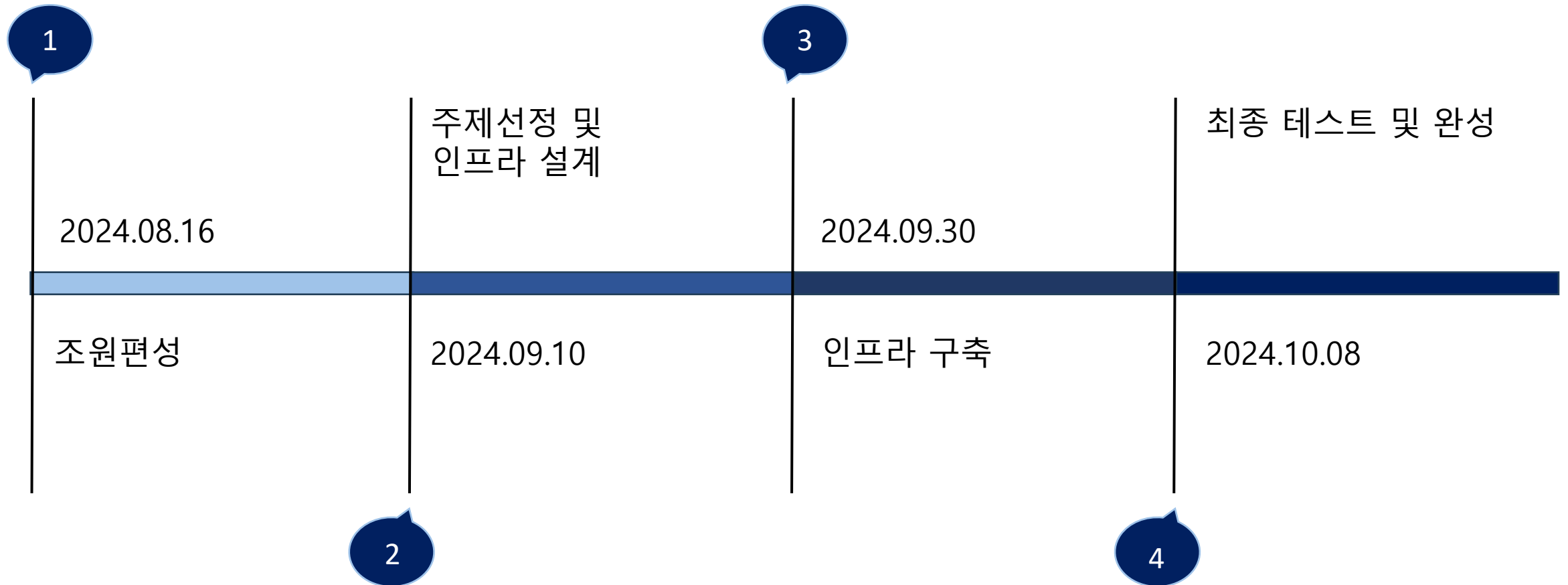
Kim jaewon



허은준

Heo eunjun

Project Schedule



□ 요리가 콘텐츠가 되는 시대

건강에 대한 관심이 증가하면서, 많은 사람들이
면역력 강화와 체중 관리를 목표로 하고 있습니다.

동시에, 배달 음식의 인기에 따른 건강 문제와 비용 부담이
커지고 있습니다.

클라우드 기반 레시피 추천 플랫폼을 구축해 개인의 목표에
맞는 맞춤형 레시피를 제공함으로써 배달 음식 의존도를 줄
이고 건강한 식습관을 형성할 수 있도록 이를 돕고자합니다.



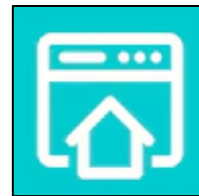
고가용성 환경 구축



데이터 보안

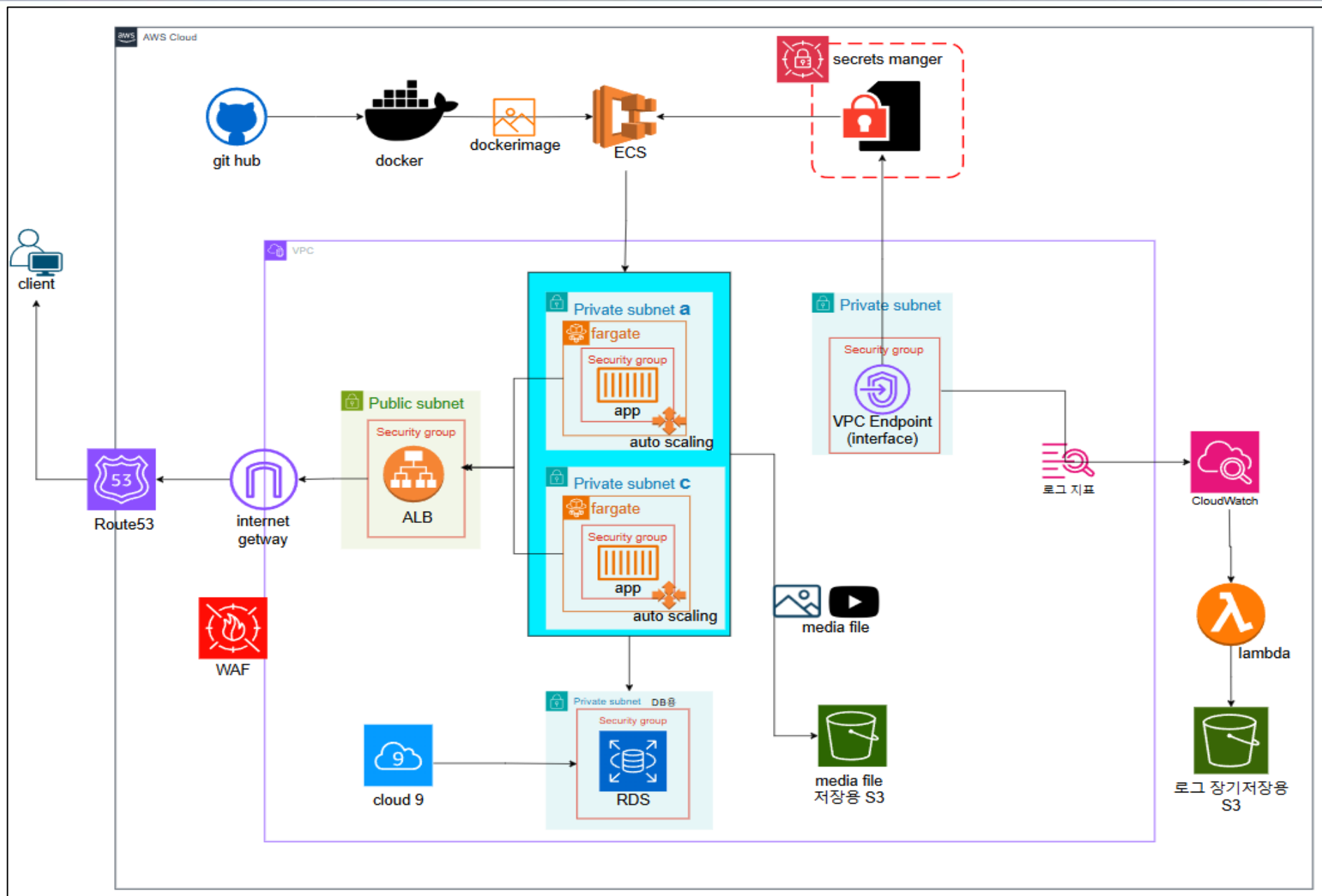


로그 분석



로그 모니터링

Architecture





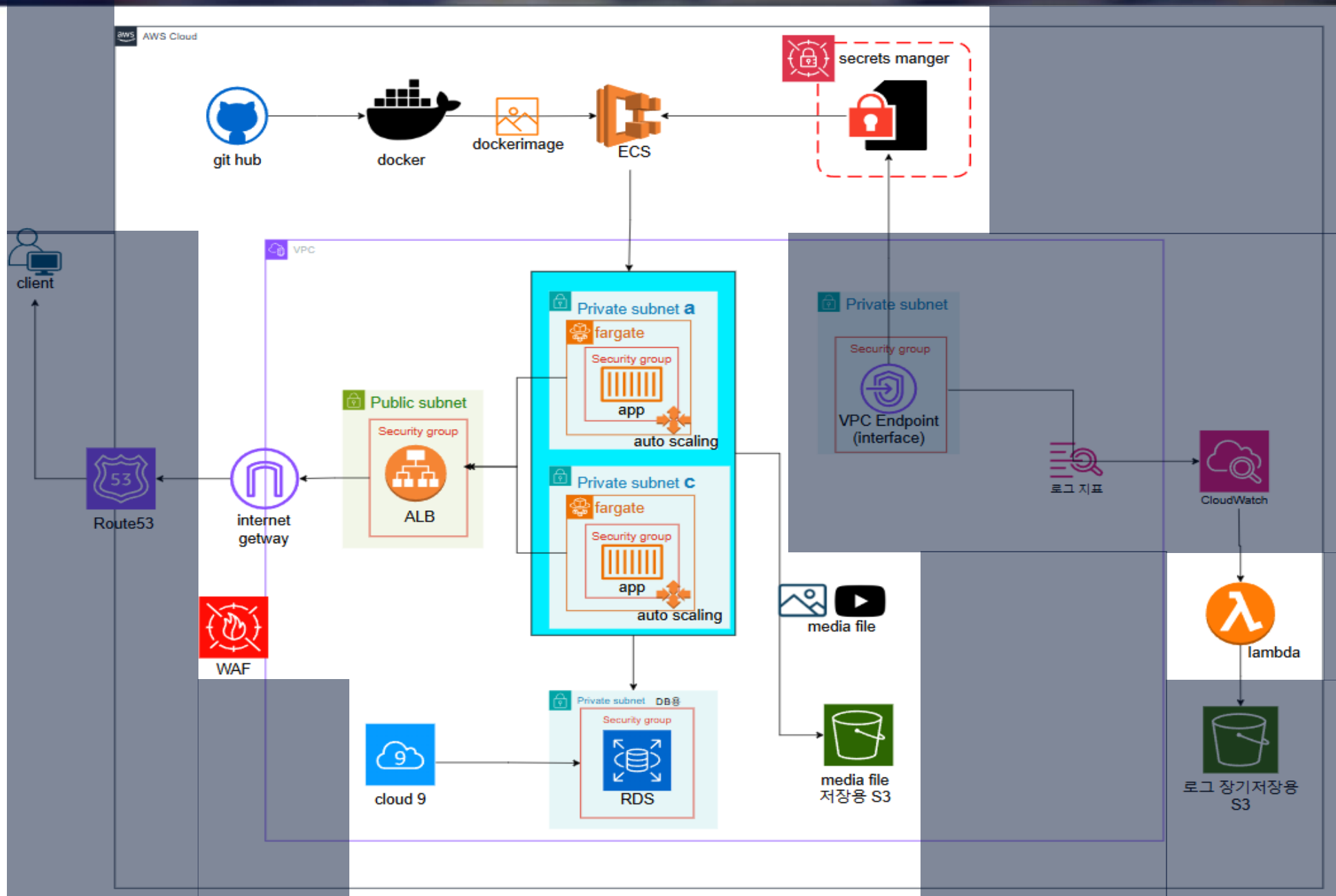
김재원 *Kim jaewon*

역할 *Role*

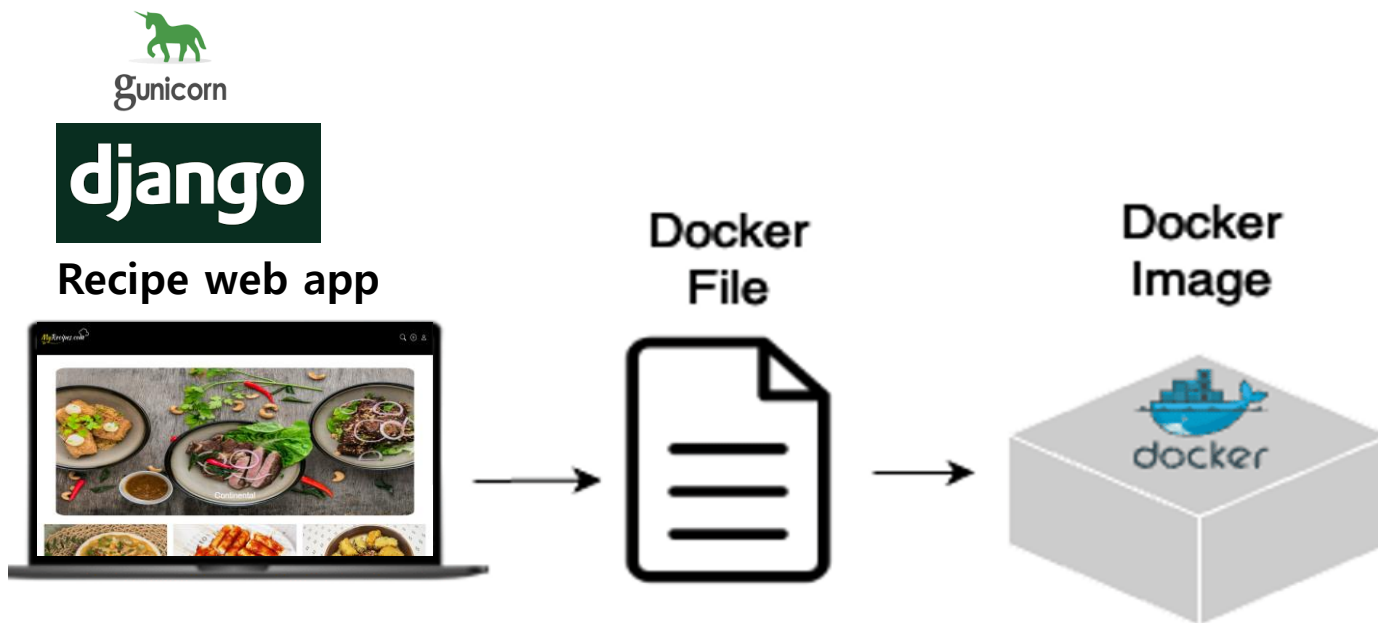
- ECS
- Secrets Manager
- RDS
- WAF
- CloudWatch
- ECR
- Lambda
- Django dev.



Architecture



- Web app 이미지 빌드



```
dockerfile  manage.py x
FROM python:3.11
WORKDIR /app
COPY . /app/
RUN pip install --upgrade pip
RUN pip install -r requirements.txt
ENV DJANGO_SETTINGS_MODULE=core.settings
```

Amazon ECR > 프라이빗 레지스트리 > 리포지토리 > epicer-repo

epicer-repo

이미지 (1)

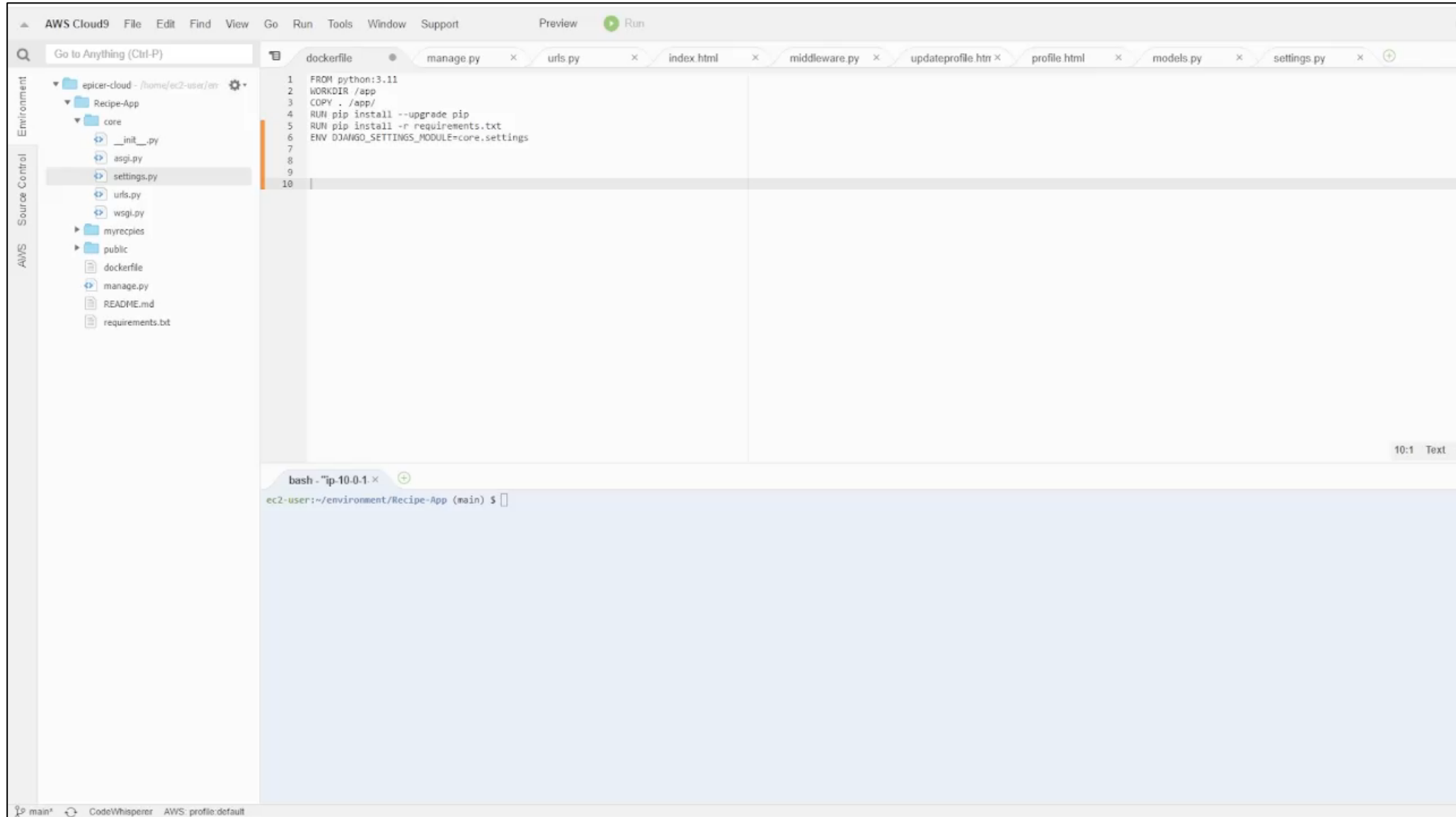
Q 아티팩트 검색

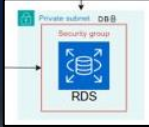
<input type="checkbox"/>	이미지 태그 ▼	아티팩트 유형	푸시 위치 ▼	크기(MB)
<input type="checkbox"/>	latest	Image	2024년 10월 09일, 09:02:08 (UTC+09)	497.76

- Dockerfile 생성
- Docker image 빌드
- ECR 이미지 푸시

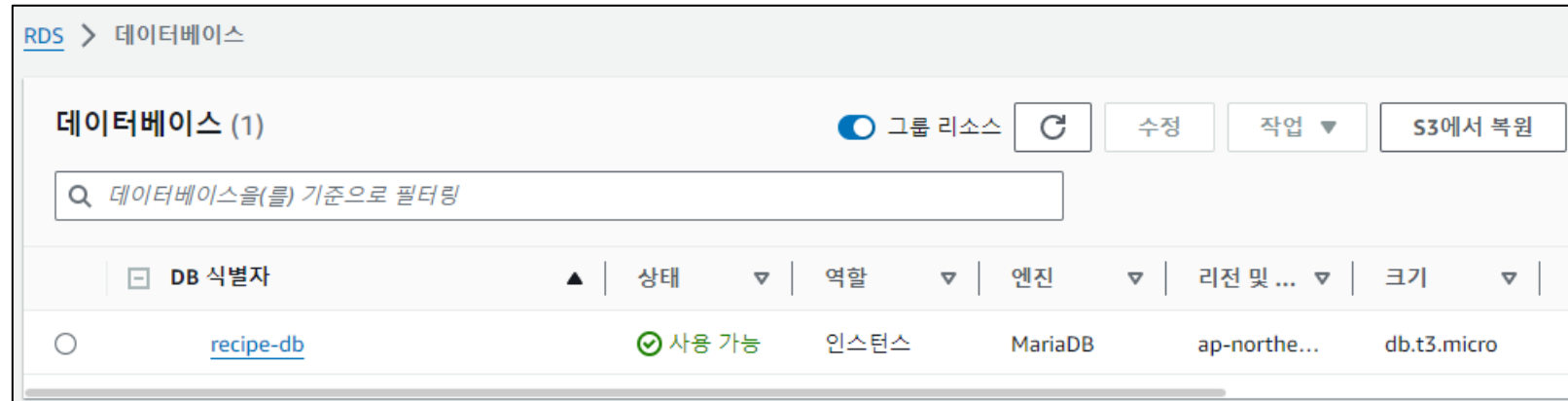
Application

- Docker image build & ECR

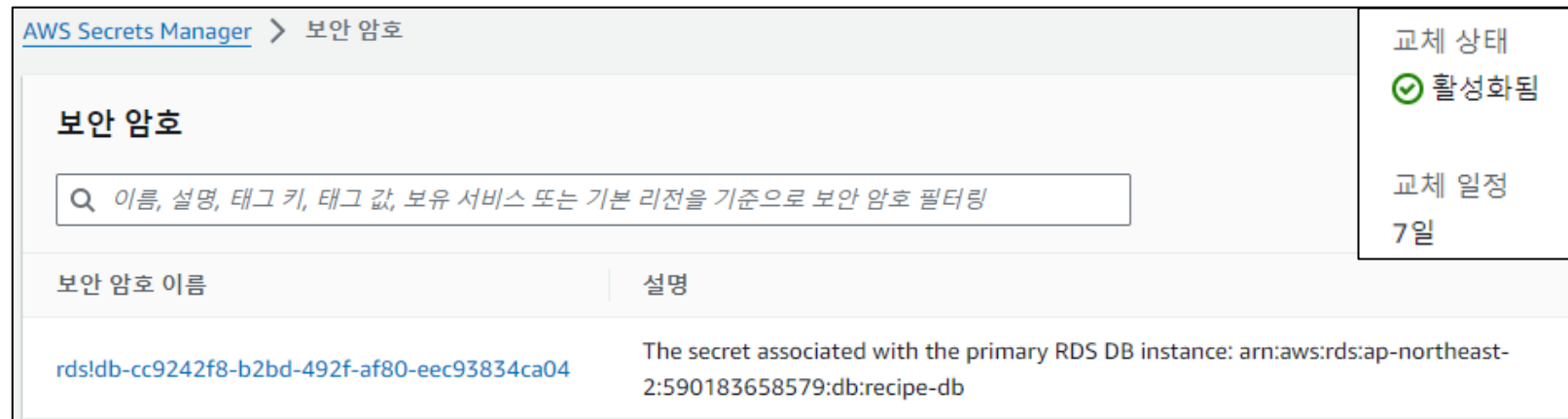




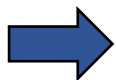
- Amazon RDS - MariaDB



트래픽량이 증가 → Amazon Aurora DB



- SQLite → Amazon RDS



```
DATABASES = {  
    'default': {  
        'ENGINE': 'django.db.backends.sqlite3',  
        'NAME': BASE_DIR / 'db.sqlite3',  
    }  
}
```



기존 소스 코드 수정
Secrets Manager 접근 권한 부여

```
def get_secret():  
  
    secret_name = "arn:aws:secretsmanager:ap-northeast-2:590183658579:secret:rds!  
    region_name = "ap-northeast-2"  
  
    session = boto3.session.Session()  
    client = session.client(  
        service_name='secretsmanager',  
        region_name=region_name  
    )  
  
    try:  
        get_secret_value_response = client.get_secret_value(SecretId=secret_name)  
    except ClientError as e:  
        raise Exception(f"Error fetching secret: {str(e)}")  
  
    secret = get_secret_value_response['SecretString']  
    return json.loads(secret)  
  
secret = get_secret()  
  
DATABASES = {  
    'default': {  
        'ENGINE': 'django.db.backends.mysql',  
        'NAME': 'epicerDB',  
        'USER': secret['username'],  
        'PASSWORD': secret['password'],  
        'HOST': 'recipe-db.cb8s888wo5vk.ap-northeast-2.rds.amazonaws.com',  
        'PORT': '3306',  
    }  
}
```

- Media file → S3

```
import os
STATIC_URL = '/static/'
STATIC_ROOT = os.path.join(BASE_DIR, 'staticfiles')

STATICFILES_DIRS = [
    os.path.join(BASE_DIR, "public/static")
]

MEDIA_ROOT = os.path.join(BASE_DIR, 'public/static')
MEDIA_URL = '/media/'
```



```
AWS_STORAGE_BUCKET_NAME = 'epicer-app-s3'
AWS_S3_REGION_NAME = 'ap-northeast-2'
AWS_S3_CUSTOM_DOMAIN = f'{AWS_STORAGE_BUCKET_NAME}.s3.amazonaws.com'

DEFAULT_FILE_STORAGE = 'storages.backends.s3boto3.S3Boto3Storage'
MEDIA_URL = f'https://{AWS_S3_CUSTOM_DOMAIN}/media/'
```









레시피 등록시 첨부 이미지 s3에 저장



Amazon S3 > 버킷 > epicer-app-s3 > media/ > recepie/




recepie/

객체 | 속성

객체 (3) 정보   S3 URI 복사  URL 복사  다운로드  열기  삭제  작업  폴더 만들기

객체는 Amazon S3에 저장되어 있는 기본 엔터티입니다. [Amazon S3 인벤토리](#)를 사용하여 버킷에 있는 모든 객체의 목록을 얻을 수 있습니다. 다른 사용자가 객체에 액세스할 수 있게 하려면 명
여해야 합니다. [자세히 알아보기](#)

🔍 접두사로 객체 찾기

<input type="checkbox"/>	이름	▲	유형	▼	마지막 수정	▼	크기	▼	스토리지 클래스
<input type="checkbox"/>	 Fried-eggplant.PNG		PNG		2024. 9. 27. am 10:46:17 AM KST		488.7KB		Standard
<input type="checkbox"/>	 gamdewitch.PNG		PNG		2024. 9. 27. am 10:46:17 AM KST		733.7KB		Standard
<input type="checkbox"/>	 ricecake-skewers.PNG		PNG		2024. 9. 27. am 10:46:18 AM KST		684.5KB		Standard



가지튀김

Created by: [99pand](#)

[View](#) [Delete](#) [Edit](#)



떡꼬치

Created by: [99pand](#)

[View](#) [Delete](#) [Edit](#)



감자달걀전

Created by: [99pand](#)

[View](#) [Delete](#) [Edit](#)

[Add More](#)

• 리스너 HTTP, HTTPS

리스너 규칙 (2) 정보 규칙 제한 🔄

리스너가 수신한 트래픽은 기본 작업 및 추가 규칙에 따라 라우팅됩니다. 규칙은 가장 낮은 값에서 가장 높은 값까지 우선 순위에 따라 평가됩니다.

🔍 규칙 필터링

<input type="checkbox"/>	이름 태그	우선 순위 ▲	조건(인 경우)	작업(다음 수행)
<input type="checkbox"/>	redirect	1	경로 패턴은 /입니다	리디렉션 대상 HTTPS://#{host}:443/#{path}?#{query} • 상태 코드: HTTP_301
<input type="checkbox"/>	기본값	마지막(기본값)	다른 규칙이 적용되지 않는 경우	대상 그룹으로 전달 • epicer-tg 1 (100%) • Target group stickiness: 끄

[AWS Certificate Manager](#) > 인증서

인증서 (2)

<input type="checkbox"/>	인증서 ID	도메인 이름 ▼	유형 ▼	상태
<input type="checkbox"/>	b01593be-d56f-473b-aec9-3a3017ddfd43	99pan.xyz	Amazon 발급	✅ 발급됨
<input type="checkbox"/>	f6ef797d-d8d6-4d80-8d65-94bb38f403ea	*.99pan.xyz	Amazon 발급	✅ 발급됨

규칙 | **인증서** | 보안 | 태그

기본 인증서: *.99pan.xyz 정보

보안 리스너를 처음 생성할 때 지정한 서버 인증서가 기본 인증서로 사용됩니다. 보안 리스너를 생성한 후 기본 인증서를 바꿀 수 있습니다.

인증서 ID f6ef797d-d8d6-4d80-8d65-94bb38f403ea	이름 또는 도메인 *.99pan.xyz	상태 ✅ 유효	SAN ..
만료 2025년 8월 21일, 08:59 (UTC+09:00)	서비스 ACM	ARN arn:aws:acm:ap-northeast-2:590183658579:certificate/f6ef797d-d8d6-4d80-8d65-94bb38f403ea	유형 Amazon 발급

- WAF

일반적인 위협을 방어
악성 IP 주소로부터의 트래픽을 차단
프록시, VPN과 같은 익명화된 IP 주소를 차단
코드 및 명령어 주입 시도와 같은 오입력을 방지
SQL 인젝션 공격을 방지

AWS WAF > Web ACLs > epicer-waf-webacl

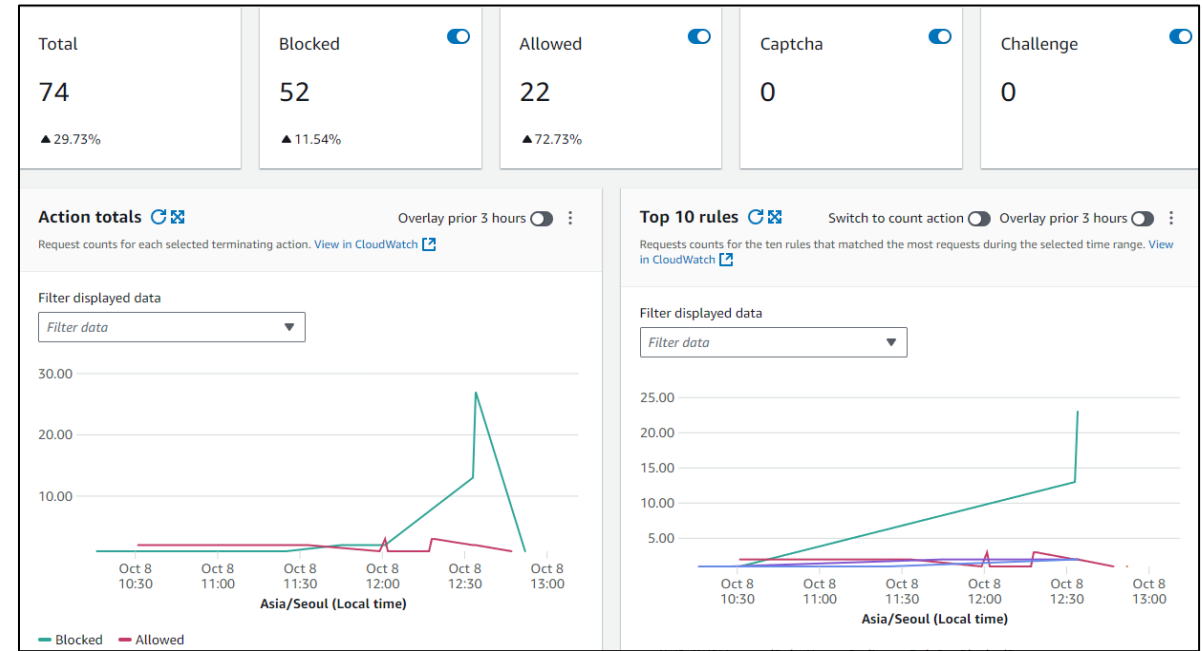
epicer-waf-webacl

Traffic overview Rules Associated AWS resources Custom response bodies Logging and metrics Sampled requests

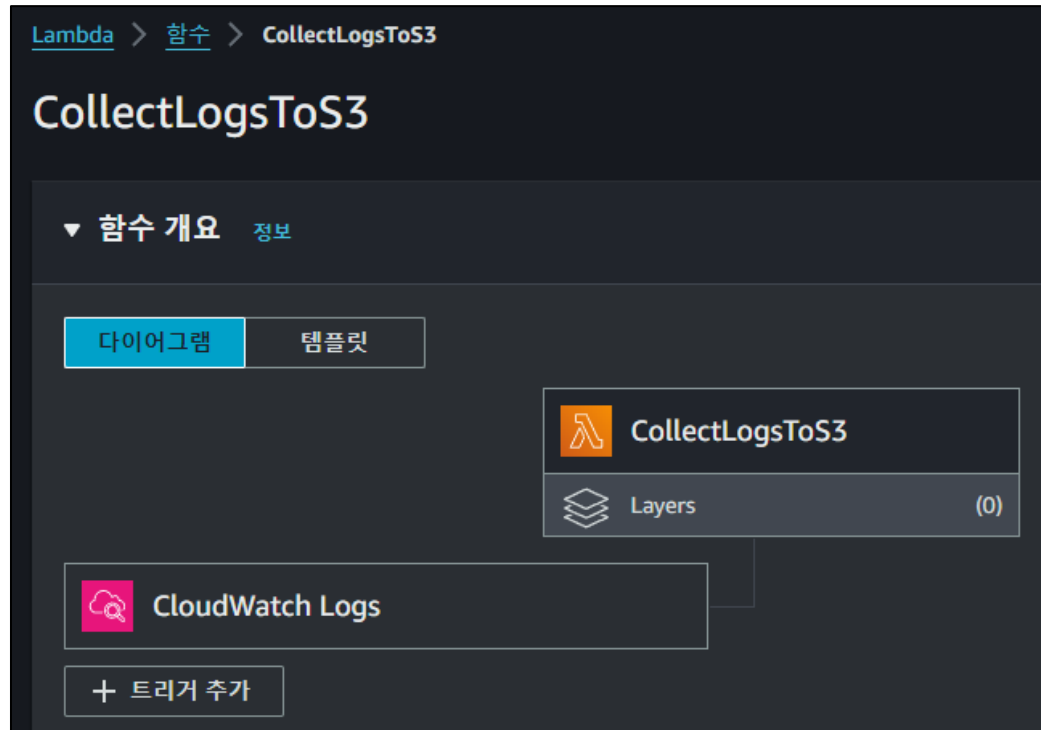
Rules (5) [Edit](#)

Find rules

<input type="checkbox"/>	Name	Action	Priority
<input type="checkbox"/>	AWS-AWSManagedRulesCommonRuleSet	Use rule actions	0
<input type="checkbox"/>	AWS-AWSManagedRulesAmazonIpReputationList	Use rule actions	1
<input type="checkbox"/>	AWS-AWSManagedRulesAnonymousIpList	Use rule actions	2
<input type="checkbox"/>	AWS-AWSManagedRulesKnownBadInputsRuleSet	Use rule actions	3
<input type="checkbox"/>	AWS-AWSManagedRulesSQLiRuleSet	Use rule actions	4



- Lambda



```
import boto3
import gzip
import json
import base64
from datetime import datetime
from io import StringIO

s3_client = boto3.client('s3')

def lambda_handler(event, context):
    try:
        log_data = base64.b64decode(event['awslogs']['data'])

        log_event = gzip.decompress(log_data).decode('utf-8')
        log_event_json = json.loads(log_event)

        log_group_name = log_event_json['logGroup']
        log_stream_name = log_event_json['logStream']

        log_events = log_event_json['logEvents']
        log_messages = [event['message'] for event in log_events]

        s3_bucket_name = 'epicer-logs'
        timestamp = datetime.now().strftime('%Y-%m-%d_%H-%M-%S')
        log_file_name = f'logs/{log_group_name}/{log_stream_name}_{timestamp}.log.gz'

        compressed_log = StringIO()
        with gzip.GzipFile(fileobj=compressed_log, mode='w') as gz_file:
            gz_file.write('\n'.join(log_messages).encode('utf-8'))

        s3_client.put_object(
            Bucket=s3_bucket_name,
            Key=log_file_name,
            Body=compressed_log.getvalue(),
            ContentType='application/gzip'
        )
```

허은준 *Heo eunjun*

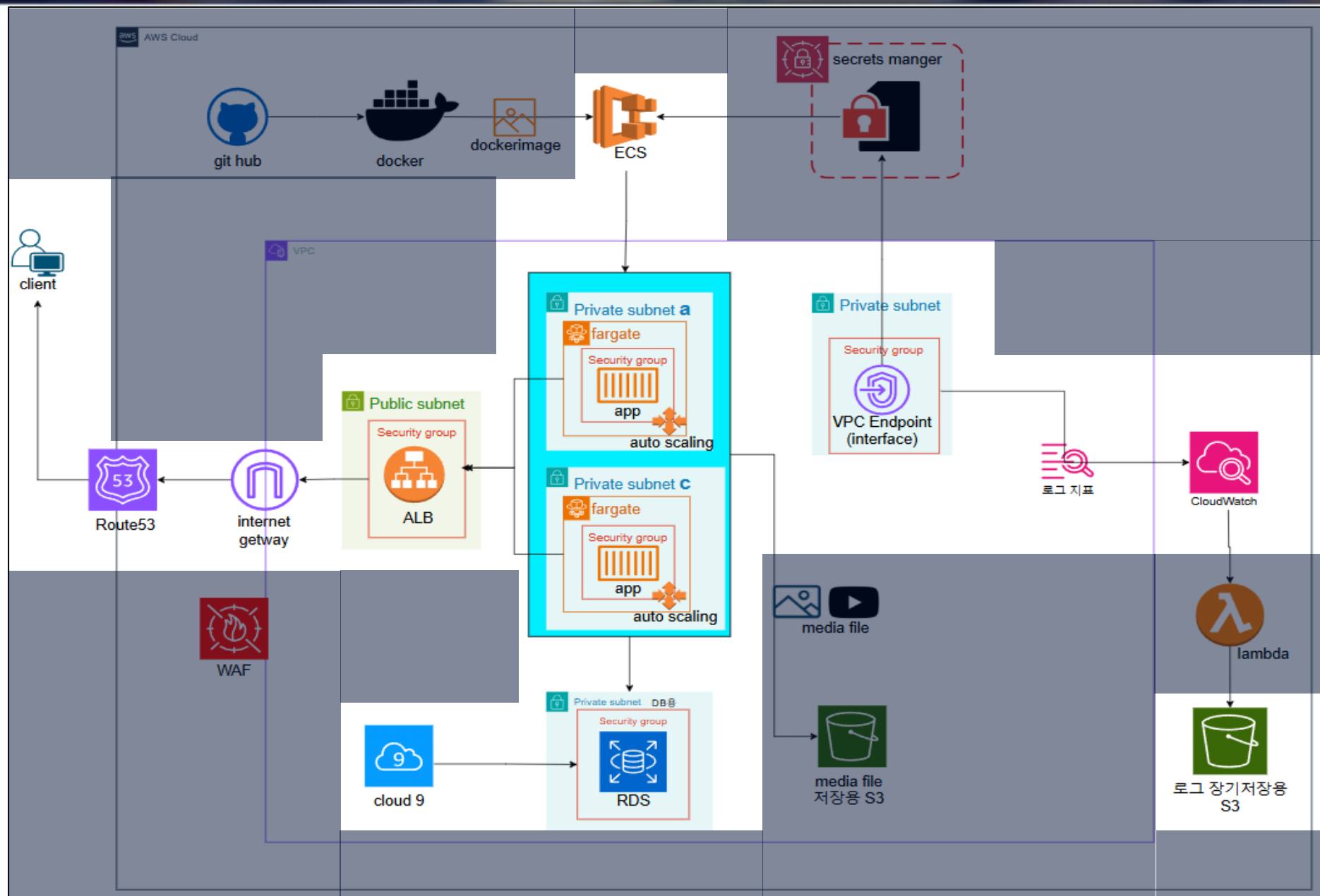


역할 *Role*

- ECS
- ECR
- RDS
- CloudWatch
- S3
- IAM
- Route53
- Terraform



Architecture



- 테라폼으로 이용한 호스팅 영역 생성

```
dns > route53.tf > ...
1 resource "aws_route53_zone" "99pan" {
2   name = "99pan.xyz"
3   comment = "Public hosted zone for 99pan.xyz"
4 }
5
6 data "aws_route53_zone" "front" {
7   name = aws_route53_zone.99pan.name
8   private_zone = false
9 }
10
11 resource "aws_acm_certificate" "cert" {
12   domain_name = "99pan.xyz"
13   validation_method = "DNS"
14
15   tags = {
16     Name = "99pan-cert"
17   }
18 }
19
20 resource "aws_route53_record" "front" {
21   zone_id = data.aws_route53_zone.front.zone_id
22   name = "99pan.xyz"
23   type = "A"
24
25   alias {
26     name = aws_lb.epicer_alb.dns_name
27     zone_id = aws_lb.epicer_alb.zone_id
28     evaluate_target_health = true
29   }
30 }
31
32 resource "aws_route53_record" "99pan_cname" {
33   zone_id = data.aws_route53_zone.front.zone_id
34   name = "www.99pan.xyz"
35   type = "CNAME"
36   ttl = 300
37   records = ["99pan.xyz"]
38 }
```

Route 53 > 호스팅 영역

호스팅 영역 (1/2)

Automatic 모드는 최상의 필터 결과에 최적화된 현재 검색 동작입니다. 모드를 변경하려면 설정(settings)으로 이동합니다.

Q 속성 또는 값을 기준으로 레코드 필터링

호스팅 영역 이름	유형	생성자	레코드 수	설명	호스팅 영역 ID
local-epicer	프라이빗	Cloud Map	2	Created by AWS Cloud	552650V8R7EN6SU
99pan.xyz	퍼블릭	Route 53	5	-	Z09161661VZDK21P92E2I

Route 53 > 호스팅 영역 > 99pan.xyz

퍼블릭 99pan.xyz 정보

▶ 호스팅 영역 세부 정보

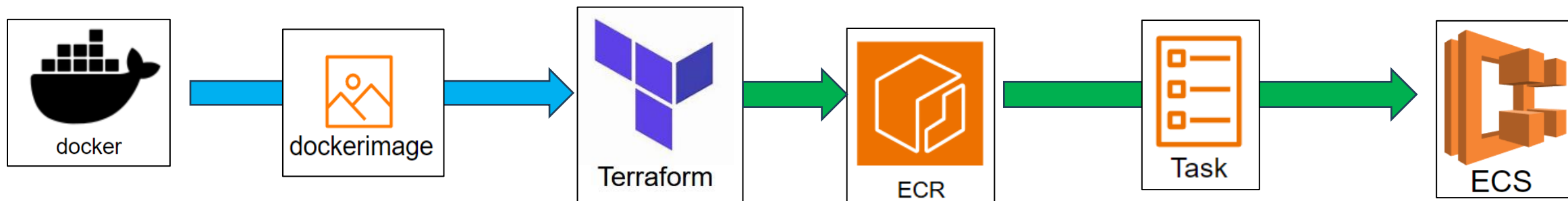
레코드(5) | DNSSEC 서명 | 호스팅 영역 태그(0)

레코드 (5) 정보

Automatic 모드는 최상의 필터 결과에 최적화된 현재 검색 동작입니다. 모드를 변경하려면 설정(settings)으로 이동합니다.

Q 속성 또는 값을 기준으로 레코드 필터링

레코드 이름	유형	라우팅 ...	차별...	별칭	값/트래픽 라우팅 대상
99pan.xyz	A	단순	-	예	dualstack.epicer-alb-ingress...
99pan.xyz	NS	단순	-	아니요	ns-948.awsdns-54.net. ns-1939.awsdns-50.co.uk. ns-171.awsdns-21.com. ns-1373.awsdns-43.org.
99pan.xyz	SOA	단순	-	아니요	ns-948.awsdns-54.net. awsd...
_340413ee88dd67301b22e7f753135fe5.99pan.xyz	CNAME	단순	-	아니요	_f22018c141b0eadc77cad92...
www.99pan.xyz	A	단순	-	예	dualstack.epicer-alb-ingress...



- Terraform 을 이용하여 클러스터 생성

Amazon Elastic Container Service > 클러스터 > epicer-ecs-app-cluster > 서비스 > epicer-app-latest > 상태

epicer-app-latest 정보

[상태 및 지표](#) | [태스크](#) | [로그](#) | [배포](#) | [이벤트](#) | [구성 및 네트워킹](#) | [태그](#)

[↻](#) [서비스 업데이트](#) [서비스 삭제](#)

상태 정보

ARN arn:aws:ecs:ap-northeast-2:590183658579:service/epicer-ecs-app-cluster/epicer-app-latest	상태 🟢 활성	태스크(2개 필요) 0개 대기 중 2개 실행 중
배포 현재 상태 🟢 2개 완료	상태 검사 유예 기간 0초	
▼ 로드 밸런서 상태 (Application Load Balancer) epicer-alb-ingress		
리스너 프로토콜:포트 HTTPS:443 🔗 HTTP:80 🔗	대상 그룹 이름:프로토콜 epicer-tg:HTTP 🔗	상태 확인 경로 /
		로드 밸런서 보기 🔗
목표 (합계 2) 🟢 2 정상 🟡 0 비정상		

- Terraform 코드를 이용하여
ECR, Task, ECS 생성

```
resource "aws_ecs_service" "my_service" {
  name           = "epicer-service"
  cluster        = aws_ecs_cluster.my_cluster.id
  task_definition = aws_ecs_task_definition.my_task.arn
  desired_count  = 2
  launch_type    = "FARGATE"

  network_configuration {
    subnets = [
      aws_subnet.SubnetPrivate1A.id,
      aws_subnet.SubnetPrivate1C.id
    ]
    security_groups = [aws_security_group.sg_private.id]
    assign_public_ip = false
  }

  load_balancer {
    target_group_arn = aws_lb_target_group.epicer_target_group.arn
    container_name   = "epicer-container"
    container_port    = 8000
  }

  tags = {
    Name = "ECS Service for Epicer App"
  }
}
```

```
1 resource "aws_ecs_cluster" "my_cluster" {
2   name = "epicer-cluster"
3 }
4
5 resource "aws_ecs_task_definition" "my_task" {
6   family           = "epicer-app-def"
7   requires_compatibilities = ["FARGATE"]
8   network_mode     = "awsvpc"
9   cpu              = "2048"
10  memory           = "6144"
11  execution_role_arn = aws_iam_role.ecs_task_execution_role.arn
12  task_role_arn     = "arn:aws:iam::590183648646:role/epicerecsTaskRole"
13  container_definitions = jsonencode([
14    {
15      name       = "epicer-container"
16      image      = "590183648646.dkr.ecr.ap-northeast-2.amazonaws.com/recipe-app-backend:latest"
17      cpu        = 0
18      memory     = 6144
19      essential  = true
20      entryPoint = ["/bin/bash", "-c"]
21      command    = ["gunicorn --workers 1 --bind 0.0.0.0:8000 --timeout 120 --log-level debug core.wsgi:application"]
22      portMappings = [
23        {
24          name       = "epicer-container-8000-tcp"
25          containerPort = 8000
26          hostPort    = 8000
27          protocol    = "tcp"
28        }
29      ]
30      logConfiguration = {
31        logDriver = "awslogs"
32        options = {
33          awslogs-group       = "/ecs/epicer-app-def"
34          awslogs-region     = "ap-northeast-2"
35          awslogs-stream-prefix = "ecs"
36          awslogs-create-group = "true"
37          mode                = "non-blocking"
38          max-buffer-size     = "25m"
39        }
40      }
41      runtimePlatform = {
42        cpuArchitecture = "X86_64"
43        operatingSystemFamily = "LINUX"
44      }
45    }
46  ])
47 }
```



로그스트림 생성 확인가능

CloudWatch > 로그 그룹 > /ecs/epicer-app-def

/ecs/epicer-app-def

▼ 로그 그룹 세부 정보

로그 클래스 : 정보

표준

ARN
arn:aws:logs:ap-northeast-2:590183648646:log-group:/ecs/epicer-app-def*

생성 시간
17시간 전

보존
언기 없음

저장된 바이트
-

자료 필터
0

구독 필터
0

가짜 사이트 규칙
-

KMS 키 ID
-

이상 탐지
구성

데이터 보호
-

민감한 데이터 개수
-

로그 스트림

태그

이상 탐지

자료 필터

구독 필터

가짜 사이트

데이터 보호

로그 스트림 (100+)

기본적으로 가장 최근의 로그 스트림만 표시합니다. [추가로 로드합니다.](#)

☐ 정확히 일치 ☐ 인접한 항목 표시 [정보](#)

<input type="checkbox"/> 로그 스트림	마지막 이벤트 시간
<input type="checkbox"/> ecs/epicer-container/861cd402b392464ab9797cbf92971020	2024-10-07 23:31:09 (UTC)
<input type="checkbox"/> ecs/epicer-container/c891f04ec388467d9434f5e8ba78afb	2024-10-07 23:28:06 (UTC)
<input type="checkbox"/> ecs/epicer-container/bf04223b58654d63b47d793853631e0b	2024-10-07 23:18:13 (UTC)
<input type="checkbox"/> ecs/epicer-container/2edc77fcc5b94a72af834b5521fc18ba	2024-10-07 23:16:47 (UTC)
<input type="checkbox"/> ecs/epicer-container/b4436f0503f94f80b602e4b3683312bb	2024-10-07 23:09:59 (UTC)
<input type="checkbox"/> ecs/epicer-container/52245991db3e43e4ab27e1a4795a3690	2024-10-07 23:07:52 (UTC)
<input type="checkbox"/> ecs/epicer-container/416b59e0add045cfb6566baec8205be1	2024-10-07 23:00:39 (UTC)
<input type="checkbox"/> ecs/epicer-container/6a270c2330d948e2ac42c73bc01b0521	2024-10-07 22:53:24 (UTC)
<input type="checkbox"/> ecs/epicer-container/7ac544a4906946eb8316bc76968d499	2024-10-07 22:46:05 (UTC)

CloudWatch > 로그 그룹 > /ecs/epicer-app-def > ecs/epicer-container/861cd402b392464ab9797cbf92971020

로그 이벤트

아래의 필터 식대를 사용하여 로그 이벤트의 줄어, 주문 또는 값을 검색하고 매칭할 수 있습니다. [필터 표현에 대해 자세히 알아보기](#)

지우기 1분 30분 1시간 12시간 사용자 지정 UTC 시간대 디스플레이

타임스탬프

메시지

▶ 2024-10-07T23:31:26.658Z

host = request.get_host()

▶ 2024-10-07T23:31:26.658Z

File "/usr/local/lib/python3.10/site-packages/django/http/request.py", line 150, in get_host

▶ 2024-10-07T23:31:26.658Z

raise DisallowedHost(msg)

▶ 2024-10-07T23:31:26.658Z

django.core.exceptions.DisallowedHost: Invalid HTTP_HOST header: '10.0.4.141:8080'. You may need to add '10.0.4.141' to ALLOWED_HOSTS.

▶ 2024-10-07T23:31:26.677Z

Bad Request: /

▶ 2024-10-07T23:31:26.677Z

[2024-10-08 05:01:26 +0530] [7] [DEBUG] GET /

▶ 2024-10-07T23:31:26.678Z

Invalid HTTP_HOST header: '10.0.4.141:8080'. You may need to add '10.0.4.141' to ALLOWED_HOSTS.

▶ 2024-10-07T23:31:26.678Z

Traceback (most recent call last):

▶ 2024-10-07T23:31:26.678Z

File "/usr/local/lib/python3.10/site-packages/django/core/handlers/exception.py", line 55, in inner

▶ 2024-10-07T23:31:26.678Z

response = get_response(request)

▶ 2024-10-07T23:31:26.678Z

File "/usr/local/lib/python3.10/site-packages/django/utils/deprecation.py", line 133, in __call__

▶ 2024-10-07T23:31:26.678Z

response = self.process_request(request)

▶ 2024-10-07T23:31:26.678Z

File "/usr/local/lib/python3.10/site-packages/django/middleware/common.py", line 48, in process_request

▶ 2024-10-07T23:31:26.678Z

host = request.get_host()

▶ 2024-10-07T23:31:26.678Z

File "/usr/local/lib/python3.10/site-packages/django/http/request.py", line 150, in get_host

▶ 2024-10-07T23:31:26.678Z

raise DisallowedHost(msg)

▶ 2024-10-07T23:31:26.678Z

django.core.exceptions.DisallowedHost: Invalid HTTP_HOST header: '10.0.4.141:8080'. You may need to add '10.0.4.141' to ALLOWED_HOSTS.

▶ 2024-10-07T23:31:26.694Z

Bad Request: /

▶ 2024-10-07T23:31:56.662Z

[2024-10-08 05:01:56 +0530] [7] [DEBUG] GET /

▶ 2024-10-07T23:31:56.662Z

Invalid HTTP_HOST header: '10.0.4.141:8080'. You may need to add '10.0.4.141' to ALLOWED_HOSTS.

▶ 2024-10-07T23:31:56.662Z

Traceback (most recent call last):

▶ 2024-10-07T23:31:56.662Z

File "/usr/local/lib/python3.10/site-packages/django/core/handlers/exception.py", line 55, in inner

▶ 2024-10-07T23:31:56.662Z

response = get_response(request)

Windows 정품 인증

[설정]으로 이동하여 Windows를 정품 인증합니다

S3 Logs



[Amazon S3](#) > [버킷](#) > [epicer-logs](#) > [logs/](#) > [/](#) > [ecs/](#) > [epicer-app-def/](#) > [ecs/](#) > [epicer-container/](#)

epicer-container/ S3 URI 복사

객체

속성

아래 표에서 정렬을 활성화하려면 검색을 사용하여 목록 크기를 999개 객체 이하로 줄이십시오.

객체 (999+) 정보

S3 URI 복사

URL 복사

다운로드

열기

삭제

작업 ▼

폴더 만들기

업로드

객체는 Amazon S3에 저장되어 있는 기본 엔터티입니다. [Amazon S3 인벤토리](#)를 사용하여 버킷에 있는 모든 객체의 목록을 얻을 수 있습니다. 다른 사용자가 객체에 액세스할 수 있게 하려면 명시적으로 권한을 부여해야 합니다. [자세히 알아보기](#)

Q 접두사로 객체 찾기

< 1 2 3 4 5 6 7 ... > ⚙

<input type="checkbox"/>	이름 ▲	유형	마지막 수정	크기	스토리지 클래스
<input type="checkbox"/>	<div><div></div><div>29f6edb7b03748c1b414c3b81b3cf542_2024-10-07_14-10-02.log.gz</div></div>	gz	2024. 10. 7. pm 11:10:04 PM KST	1.3KB	Standard
<input type="checkbox"/>	<div><div></div><div>d7034fb2a92945b99cfe073f08bbd52d_2024-10-03_07-38-57.log.gz</div></div>	gz	2024. 10. 3. pm 4:38:58 PM KST	68.0B	Standard
<input type="checkbox"/>	<div><div></div><div>d7034fb2a92945b99cfe073f08bbd52d_2024-10-03_07-39-12.log.gz</div></div>	gz	2024. 10. 3. pm 4:39:13 PM KST	68.0B	Standard
<input type="checkbox"/>	<div><div></div><div>d7034fb2a92945b99cfe073f08bbd52d_2024-10-03_07-39-27.log.gz</div></div>	gz	2024. 10. 3. pm 4:39:28 PM KST	68.0B	Standard
<input type="checkbox"/>	<div><div></div><div>d7034fb2a92945b99cfe073f08bbd5</div></div>	gz	2024. 10. 3. pm 4:39:43 PM KST	66.0B	Standard

Terraform 서비스 구현

The screenshot displays a Windows desktop environment. The main window is a text editor showing a Terraform configuration file named `ecs.tf`. The file contains the following content:

```
1 # ECS 클러스터 생성
2 resource "aws_ecs_cluster" "my_cluster" {
3   name = "epicer-cluster"
4 }
5
6 # ECS 태스크 정의 생성 (Fargate)
7 resource "aws_ecs_task_definition" "my_task" {
8   family           = "epicer-app-def"
9   requires_compatibilities = ["FARGATE"]
10  network_mode     = "awsvpc"
11  cpu              = "2048"
12  memory           = "6144"
13  execution_role_arn = aws_iam_role.ecs_task_execution_role.arn
14  task_role_arn     = "arn:aws:iam::590183648646:role/epicerTaskRole" # 태스크 역할 수정
15  container_definitions = jsonencode([
16    {
17      name       = "epicer-container"
18      image      = "590183648646.dkr.ecr.ap-northeast-2.amazonaws.com/recipe-app-backend:latest" # ECR 이미지 수정
19      cpu        = 0
20      memory     = 6144
21      essential  = true
22      entryPoint = ["/bin/bash", "-c"]
23      command    = ["gunicorn --workers 1 --bind 0.0.0.0:8000 --timeout 120 --log-level debug core.wsgi:application"]
24      portMappings = [
25        {
26          containerPort = 8000
27          hostPort      = 8000
28        }
29      ]
30    }
31  ])
32 }
```

Below the text editor, a terminal window is open, showing the command `terraform apply` being executed. The terminal output is as follows:

```
C:\Users\soldesk\Desktop\reend\dns>terraform apply
[...]
```

The Windows taskbar at the bottom shows the system clock as 12:45 on 16th of the month. A Windows watermark is visible in the bottom right corner.

아쉬운점

김재원

로그를 이용한 시각화 도구 그래파나와 클라우드 모니터링 프로그램 데이터독을 사용해보지 못한 것이 아쉽습니다.

허은준

프로젝트 도중 결원이 생긴상태로 진행하여 더 많은 서비스를 해보지 못한 것
람다를 테라폼으로도 시도했지만 시간부족으로 못한 것이 너무 아쉽습니다.

A man in a grey suit and tie is blindfolded with a black cloth. He is screaming with his mouth wide open. Another man in a dark suit is feeding him with a spoon. The man being fed is holding a white bowl. The background is dark and textured.

Q&A

크게 벌려 주세요