

아파치 서버 자동 재시작

이재우

INDEX

01 AWS

- AWS란?
- 특징

02 서버안내

- IAM이란?
- Lambda란?
- EventBridge란?

03 아파치 서버 자동 재시작

- IAM 정책, 역할 만들기
- Lambda 만들기
- EventBridge 만들기

04 소감 및 계획

- 공부 계획
- 느낀점

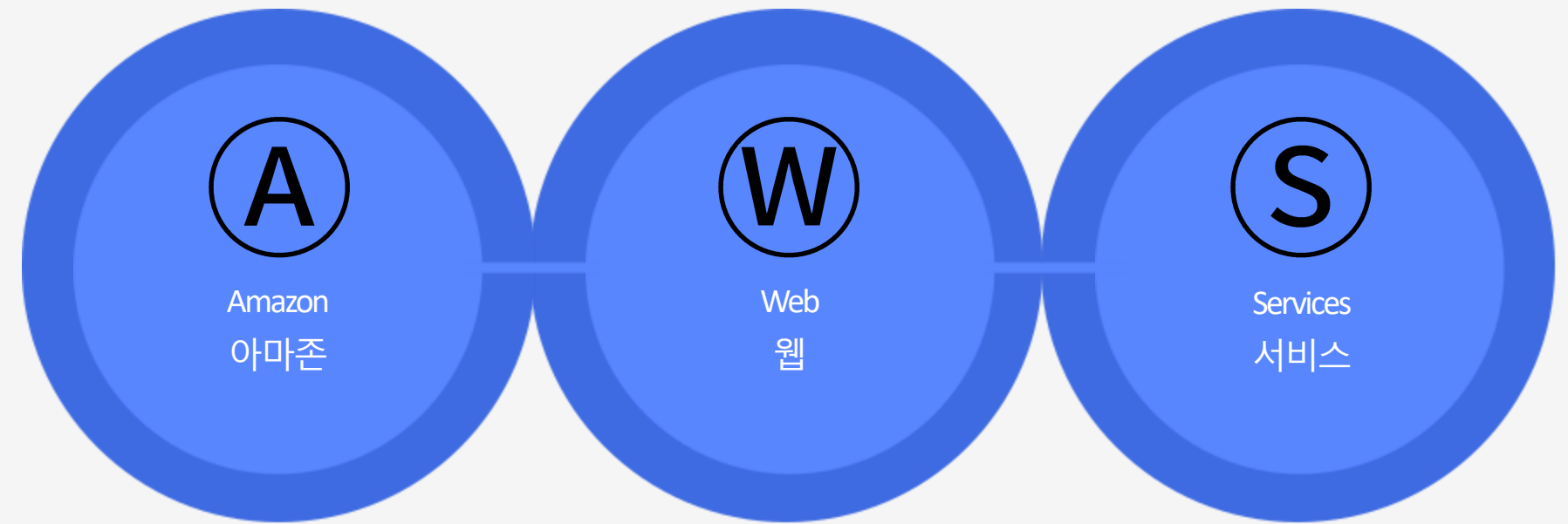


01

AWS란?

- AWS란?
- 특징

AWS란?



ChatGPT

AWS는 아마존이 제공하는 클라우드 컴퓨팅 서비스로, 기업이나 개인이 인터넷을 통해 다양한 IT 리소스를 확장하고 운영할 수 있게 해주는 플랫폼입니다. 이를 통해 컴퓨팅 파워, 데이터베이스 스토리지, 보안 등을 제공하여 유연하고 확장 가능한 클라우드 솔루션을 제공합니다.



특징



- 서버, 네트워크 등 인프라 전체를 빌려주는 서비스
- 다양한 서비스를 제공하고 있어서 AWS가 제공하는 서비스만으로 필요한 기능을 대부분 구축 가능
- 대표적인 서비스 : Amazon EC2, Amazon S3

02

사용할 서비스

- IAM이란?
- Lambda란?
- EventBridge란?

IAM이란?



- IAM은 AWS 서비스에 대한 인증 및 권한 부여를 제공합니다.
- 서비스는 AWS 요청이 허용되거나 거부되는지 평가합니다.
- 액세스는 기본적으로 거부되며 정책에서 명시적으로 액세스 권한을 부여하는 경우에만 허용됩니다.

IAM이란?

-IAM은 AWS 서비스에 대한 인증 및 권한 부여를 관리하는 서비스입니다.

AWS IAM은 Amazon Web Services(AWS)의 리소스에 접근할 수 있는 사용자와 그들의 권한을 관리하는 서비스입니다. 사용자는 필요한 권한을 가진 정책을 부여받아 AWS 서비스를 안전하게 이용할 수 있습니다.

...에서 명시적으로 액세스 권한을 부여하는 경우에만 허용됩니다.

Lambda란?



- AWS Lambda는 서버를 프로비저닝하거나 관리하지 않고도 코드를 실행할 수 있게 해주는 컴퓨팅 서비스입니다.
- 자동 확장: Lambda 함수는 요청에 따라 자동으로 확장됩니다. 요청 수가 증가하면 Lambda가 자동으로 더 많은 리소스를 할당하고, 요청 수가 감소하면 리소스를 줄여 비용을 절감합니다.

Lambda란?

lambda를 어떻게 이해하면 편할까?



AWS Lambda를 이해하기 쉽게 비유하자면, **자동조리기로** 생각할 수 있습니다. 사용자가 요리 재료(코드)를 준비해놓으면, 자동조리기(Lambda)가 필요한 시점에 알아서 조리를 시작하고(코드 실행), 필요한 만큼 재료를 추가해주고(자동 확장), 조리가 끝나면 자동으로 꺼집니다(리소스 관리). 사용자는 요리(코드)만 준비하면 되고, 나머지는 자동조리기가 알아서 해주는 것입니다. 이처럼 Lambda는 인프라 관리를 신경 쓰지 않고 코드 실행에만 집중할 수 있게 해줍니다.

EventBridge란?



- EventBridge는 어플리케이션 간의 통합을 단순화하고 이벤트 기반 아키텍처를 구현하는 데 도움을 줍니다.
- 규칙 설정: 이벤트를 특정 조건에 따라 필터링하고, 조건에 맞는 이벤트만 특정 대상으로 전달하는 규칙을 설정할 수 있습니다.
JSON 패턴 매칭을 통해 이루어집니다.
- 대상은 AWS Lambda 함수, SQS 큐, SNS 토픽, 등 다양합니다.



03

아파치 서버 자동 재시작

- IAM 정책, 역할 만들기
- Lambda 만들기
- EventBridge 만들기

참고: 다음 단계를 완료한 후에 **Client error on launch** 오류가 발생할 수 있습니다. 자세한 내용을 보려면, 암호화된 볼륨이 연결된 상태로 인스턴스를 시작하면 "client error on launch"라는 오류와 함께 인스턴스가 즉시 중지됩니다를 참조하세요.

중지 및 시작하려는 EC2 인스턴스의 ID를 가져옵니다. 이어서, 다음 단계를 완료합니다.

Lambda 함수에 대한 IAM 정책 및 IAM 역할 생성

- 1. JSON 정책 편집기를 사용하여 IAM 정책을 생성합니다. 다음 JSON 정책 문서를 정책 편집기에 붙여 넣습니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ],
      "Resource": "arn:aws:logs:*:*:*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:Start*",
        "ec2:Stop*"
      ],
      "Resource": "*"
    }
  ]
}
```

- 2. Lambda에 대한 IAM 역할을 생성합니다.

중요: 권한 정책을 Lambda에 연결할 때 해당 IAM 정책을 선택해야 합니다.

참고: 고객 관리형 AWS Key Management Service(AWS KMS) 키로 암호화된 Amazon Elastic Block Store(Amazon EBS) 볼륨을 사용하는 경우...

AWS re:Post을(를) 사용하면 다음에 동의하게 됩니다. AWS re:Post [이용 약관](#)



질문, 지식 센터, 기사, 주제, 사용자 등을 검색하세요.

한국어 리소스 로그인

홈 질문 지식 센터 커뮤니티 글 셀렉션 태그 더...

질문하기

인스턴스를 중지 및 시작하는 Lambda 함수 생성

1. Lambda 콘솔을 연 다음, 함수 생성을 선택합니다.
2. 새로 작성을 선택합니다.
3. 기본 정보에 다음 정보를 입력합니다.
함수 이름에는 함수를 설명하는 이름(예: "StopEC2Instances")을 입력합니다.
런타임은 Python 3.9를 선택합니다.
권한은 기본 실행 역할 변경을 확장합니다.
실행 역할은 기존 역할 사용을 선택합니다.
기존 역할에서 해당 IAM 역할을 선택합니다.
4. 함수 생성을 선택합니다.
5. 코드 탭의 코드 소스에서 다음 코드를 lambda_function 탭의 코드 편집기에 있는 편집기 창에 붙여 넣습니다. 이 코드는 다음과 같이 사용자가 식별한 인스턴스를 중지합니다.

```
import boto3
region = 'us-west-1'
instances = ['i-12345cb6de4f78g9h', 'i-08ce9b2d7eccf6d26']
ec2 = boto3.client('ec2', region_name=region)

def lambda_handler(event, context):
    ec2.stop_instances(InstanceIds=instances)
    print('Stopped your instances: {}'.format(instances))
```

피드백

상단

함수 StopServer이(가) 생성되었습니다. 이제 함수의 코드 및 구성을 변경할 수 있습니다. 테스트 이벤트를 사용하여 함수를 호출하려면 [테스트]를 선택하십시오.

코드 테스트 모니터링 구성 별칭 버전

코드 소스 정보

에서 업로드

File Edit Find View Go Tools Window Test Deploy Changes not deployed

Go to Anything (Ctrl-P)

lambda_function x Environment Vari x

Environment StopServer - / lambda_function.py

```
1 import boto3
2 region = 'ap-northeast-2'
3 instances = ['i-0e9aa559497c2e6ff']
4 ec2 = boto3.client('ec2', region_name=region)
5
6 def lambda_handler(event, context):
7     ec2.stop_instances(InstanceIds=instances)
8     print('stopped your instances: ' + str(instances))
```

```
import boto3
region = 'ap-northeast-2'
instances = ['아까 복사한 인스턴스ID']
ec2 = boto3.client('ec2', region_name=region)

def lambda_handler(event, context):
    ec2.stop_instances(InstanceIds=instances)
    print('stopped your instances: ' + str(instances))
```

3:34 Python Spaces: 4

코드 손성 정보

함수 StopServer이(가) 생성되었습니다. 이제 함수의 코드 및 구성을 변경할 수 있습니다. 테스트 이벤트를 사용하여 함수를 호출하려면 [테스트]를 선택하십시오.

코드 테스트 모니터링 구성 별칭 버전

코드 소스 정보

에서 업로드

File Edit Find View Go Tools Window Test Deploy Changes not deployed

Go to Anything (Ctrl-P)

lambda_function x Environment Vari x

Environment StopServer - / lambda_function.py

```
1 import boto3
2 region = 'ap-northeast-2'
3 instances = ['i-0e9aa559497c2e6ff']
4 ec2 = boto3.client('ec2', region_name=region)
5
6 def lambda_handler(event, context):
7     ec2.stop_instances(InstanceIds=instances)
8     print('stopped your instances: ' + str(instances))
```

```
import boto3
region = 'ap-northeast-2'
instances = ['인스턴스ID']
ec2 = boto3.client('ec2', region_name=region)

def lambda_handler(event, context):
    ec2.start_instances(InstanceIds=instances)
    print('started your instances: ' + str(instances))
```

3:34 Python Spaces: 4

코드 손성 정보

발생 정보

일회성 또는 반복 일정을 정의할 수 있습니다.

일회성 일정 반복 일정

시간대

일정의 시간대입니다.

(UTC+09:00) Asia/Seoul

일정 유형

요구 사항에 가장 잘 맞는 일정 유형을 선택합니다.

Cron 기반 일정
매월 첫 번째 월요일 오전 8시(PST)와 같이 특정 시간에 실행되는 Cron 표현식을 사용하여 설정된 일정입니다.
 Rate 기반 일정
10분마다와 같이 일정한 빈도로 실행되는 일정입니다.

Cron 표현식 정보

일정에 대한 cron 표현식 정의

복사 | 지우기

cron (0 7 * * ? *)
분 시간 일 월 요일 연도

다음 10개의 트리거 날짜

날짜와 시간은 태평양 표준시의 경우 '2022년 11월 9일 수요일 09:00(UTC - 08:00)'처럼 현재 시간대를 기준으로 UTC 형식으로 표시됩니다

- Tue, 14 May 2024 07:00:00 (UTC+09:00)
- Wed, 15 May 2024 07:00:00 (UTC+09:00)
- Thu, 16 May 2024 07:00:00 (UTC+09:00)
- Fri, 17 May 2024 07:00:00 (UTC+09:00)
- Sat, 18 May 2024 07:00:00 (UTC+09:00)
- Sun, 19 May 2024 07:00:00 (UTC+09:00)
- Mon, 20 May 2024 07:00:00 (UTC+09:00)
- Tue, 21 May 2024 07:00:00 (UTC+09:00)
- Wed, 22 May 2024 07:00:00 (UTC+09:00)
- Thu, 23 May 2024 07:00:00 (UTC+09:00)

유연한 기간

유연한 기간을 선택하면 Scheduler는 지정한 시간 범위 내에서 일정을 호출합니다. 예를 들어 15분을 선택하면 스케줄이 시작 시간으로부터 15분 이내에 실행됩니다.

꺼짐

504 Gateway Time-out

504 Gateway Time-out

공지사항 | 갤러리 | 공모전 알림 | 자유게시판 | 자료실 | 전공동아리 | 수상현황 | 보안뉴스 | 포털연혁 | 실습실A/S | 유머 페이지

널포유 개발용2 - 세명검고 보안과 포털

SMC SECU PORTAL

공지사항 | 갤러리 | 공모전 알림 | 자유게시판 | 자료실 | 전공동아리 | 수상현황 | 보안뉴스 | 포털연혁 | 실습실A/S | 유머 페이지

보안과 사제 교류 친선 축구게임

By admin | March 19, 2024 | 공지사항 [Leave a comment](#)



세레머니는 월클ㅋㅋㅋ

이번 주 금요일
사제 교류 축구 게임 개시!

일시 : 3/22(금) 16:00 ~ 18:00
장소 : 진관풋살구장(신도고 뒤 물푸레근린공원)

< May 2024 >

S	M	T	W	T	F	S
28	29	30	1	2	3	4
1차 지필평가					체육대	
5	6	7	8	9	10	11
		3일차				
12	13	14	15	16	17	18
				리더십 캠프		
19	20	21	22	23	24	25
26	27	28	29	30	31	1
			수락여행			

Search

2023 대입 BoB(Best Of Best) CTF
CTF 풀이 GoPRO Null2U SCA 교내CTF 널포유

농구 대입 대회 동아리

매뉴얼 방과후 방과후신청 사용법
사이버가디언즈 사제교류 서클여대 수기
수시 스마트보안솔루션 경진대회 스카
신인보원모집 신청 방법 동순장품

04

소감 및 계획

- 문제 원인 찾기
- 느낀점

문제

1. **서버 로그 확인:** 먼저 아파치 웹 서버의 로그 파일을 확인하여 다운되는 시기와 원인을 파악합니다. 로그 파일에는 주요한 에러 메시지나 경고가 기록될 수 있습니다.
2. **시스템 리소스 확인:** 서버가 다운되는 동안 CPU, 메모리 및 네트워크 리소스를 확인하여 서버 부하가 문제의 원인인지 확인합니다.
3. **네트워크 연결 및 환경:** 서버와 데이터베이스 간의 네트워크 연결이 안정적인지 확인하고, 환경 변수나 설정이 문제를 일으키는지 확인합니다.
4. **서버 구성 검토:** 아파치 웹 서버의 구성 파일을 검토하여 문제가 있는지 확인합니다. 특히, 서버 설정이나 모듈 설정에 오류가 있는지 확인합니다.
5. **서버 모니터링 도구 사용:** 서버의 상태를 지속적으로 모니터링하고, 다운되는 시기나 원인을 추적할 수 있는 모니터링 도구를 사용합니다.
6. **버전 및 패치 확인:** 아파치 웹 서버와 관련된 소프트웨어의 버전 및 패치를 확인하여 최신 버전으로 업데이트되어 있는지 확인합니다.
7. **외부 요인 확인:** 외부 공격, 네트워크 문제 또는 다른 시스템의 문제로 인해 서버가 다운되는 경우도 있으므로 외부 요인을 고려합니다.

느낌점

