



Gurumi Bot with RAG

- Amazon Bedrock
- Knowledge bases
- OpenSearch

날밤

(유정열, nalbam)

당근SRE팀

AWS Machine Learning Hero



DeepRacer Timer

 [nalbam's blog](#)

About   

DeepRacer Timer and Sensor Production Log

NOVEMBER 07, 2019



I participated AWS re:Invent 2018 last year. AWS presented Deepracer, I participated in the workshop and received a DeepRacer. From then on, I trained the DeepRacer, participated in some leagues, and achieved good grades. I won the 9th place at the Seoul Summit and 22nd place at the Tokyo Summit... And I will be able to take part in the finals at AWS re:Invent 2019 since I won the 6th place in the total points of 6 online league seasons.

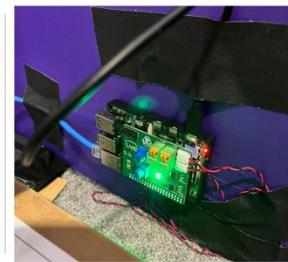
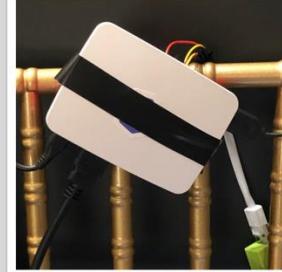
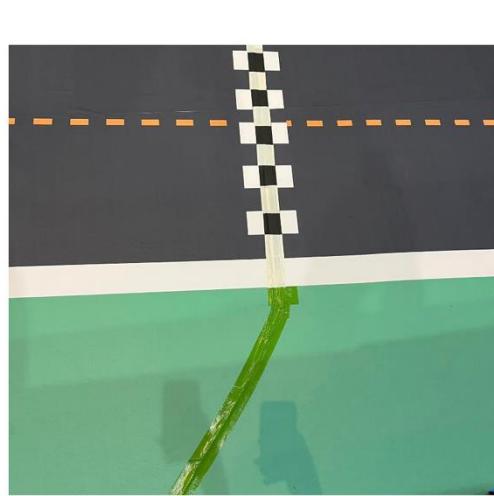
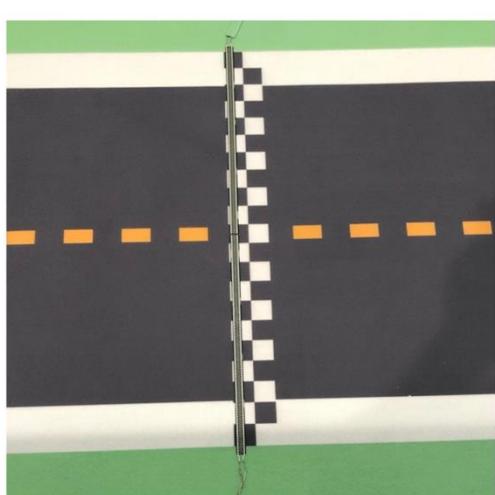
5/31	Karl-NAB	9.739	
6/30	Fumiaki	10.31	
7/31	2edgy	9.203	
8/31	Breadcentric	8.03	
9/31	JasonLion	8.72	
10/31	Eric-NCTU-CGI	7.77	
V1	Etaggel	5938.23	
V2	pgs-tomasz-ponek	5937.82	
V3	Nero-DNPOS	5937.28	
V4	oiiis-dnp	5934.19	
V5	jimwu	5933.78	
V6	Nalbam-me	5933.20	
v7	kimwooglae	5932.70	
v8	jochem	5931.39	
v9	obaykov	5930.48	
v10	hiroisojp	5928.38	
v11	leo-dnpps	5929.13	
v12	mogamin	5927.81	
v13	kire	5926.78	
v14	hyeonwoo	5925.88	
v15	tonyj	5925.77	
v16	richardfan	5925.17	
v17	robin-castro	5920.27	
v18	Kagrazaka-DNP	5917.02	



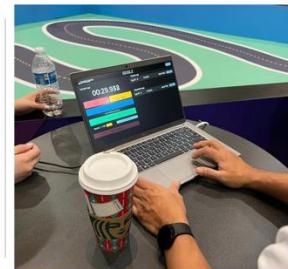
A QR code is located on the right side of the page.



DeepRacer Timer



2019 | 2022



2019 | 2022



ChatGPT Slack Bot with ChatGPT

유정열 / 당근마켓 SRE팀, AWS ML Hero

2023-06-07

ChatGPT Slack Bot



그래서 물어보면?



nalbam 오늘 오전 10:41

@Gurumi (구루미) 딥레이서 타이머가 뭐야



Gurumi (구루미) 앱 오늘 오전 10:41

딥레이서 타이머는 AWS DeepRacer 대회나 경주에서 사용되는 시간 측정 도구입니다. AWS DeepRacer는 AWS에서 제공하는 자율주행 자동차 레이싱 플랫폼으로, 강화학습을 이용해 모델을 훈련시키고 경주를 합니다.

딥레이서 타이머의 주요 기능은 다음과 같습니다:

1. 랩 타임 측정: 자동차가 트랙 한 바퀴를 완주하는 데 걸리는 시간을 정확히 측정합니다.
2. 경주 시작과 종료 시간 기록: 경주의 시작과 끝을 정확히 기록합니다.
3. 실시간 기록 표시: 현재 진행 중인 랩 타임과 최고 기록을 실시간으로 보여줍니다.
4. 기록 저장 및 분석: 측정된 시간 데이터를 저장하고, 이를 바탕으로 성능 분석을 할 수 있게 해줍니다.

이 타이머는 공정한 경쟁과 정확한 성능 평가를 위해 중요한 역할을 합니다. 실제 AWS DeepRacer 리그나 대회에서는 공식 타이머 시스템을 사용하여 정확성과 일관성을 보장합니다.

다만, 정확한 사양이나 세부 기능에 대해서는 AWS의 공식 문서나 최신 정보를 확인하는 것이 좋습니다. AWS DeepRacer 서비스는 지속적으로 업데이트되고 있어 새로운 기능이 추가될 수 있기 때문입니다. (편집됨)

학습이 끝난 LLM 챗봇

- 학습 당시의 데이터를 기반으로 답변
- 최신의 데이터를 제공 하지 못함
- 공개되지 않은 데이터를 제공 하지 못함
- 내가 만든 문서는 알지 못함

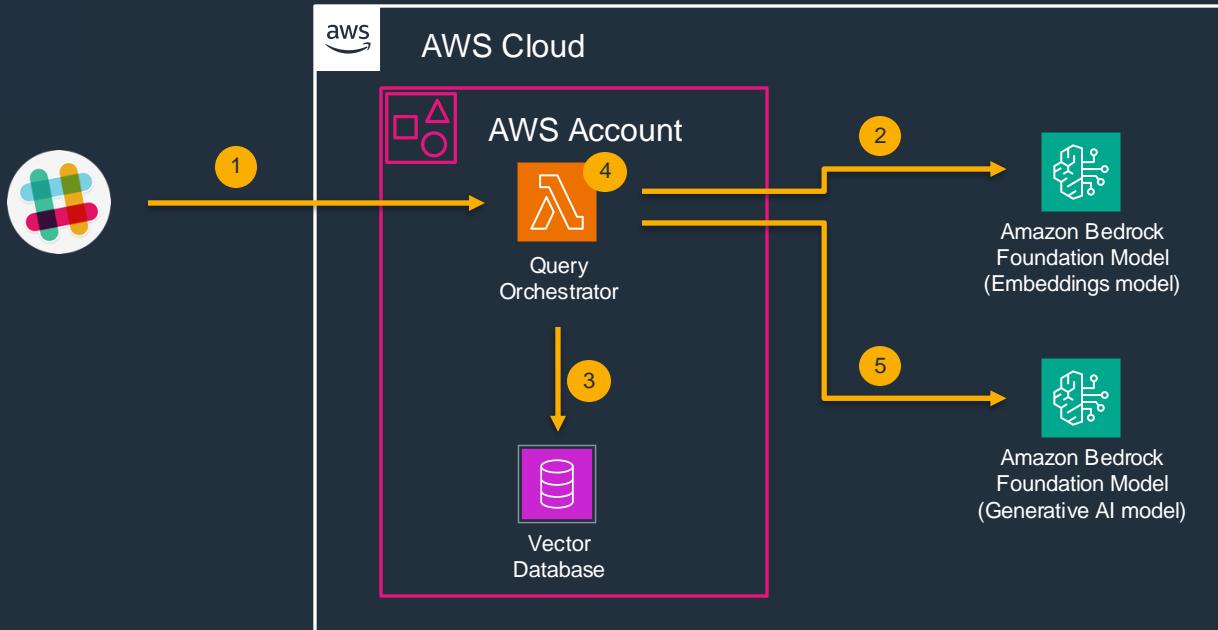
RAG ?

- Retrieval-Augmented Generation (RAG, 검색 증강 생성)
- 정보 검색 단계 (Retrieval):
 - 입력된 질문이나 문장에 대해 관련 있는 문서나 텍스트 조각을 검색
 - ex: Knowledge Bases for Amazon Bedrock
- 텍스트 생성 단계 (Generation):
 - 검색된 정보를 바탕으로 언어 모델이 응답을 생성
 - ex: Amazon Bedrock Foundation Model – Claude 3.5 Sonnet

Knowledge Bases for Amazon Bedrock

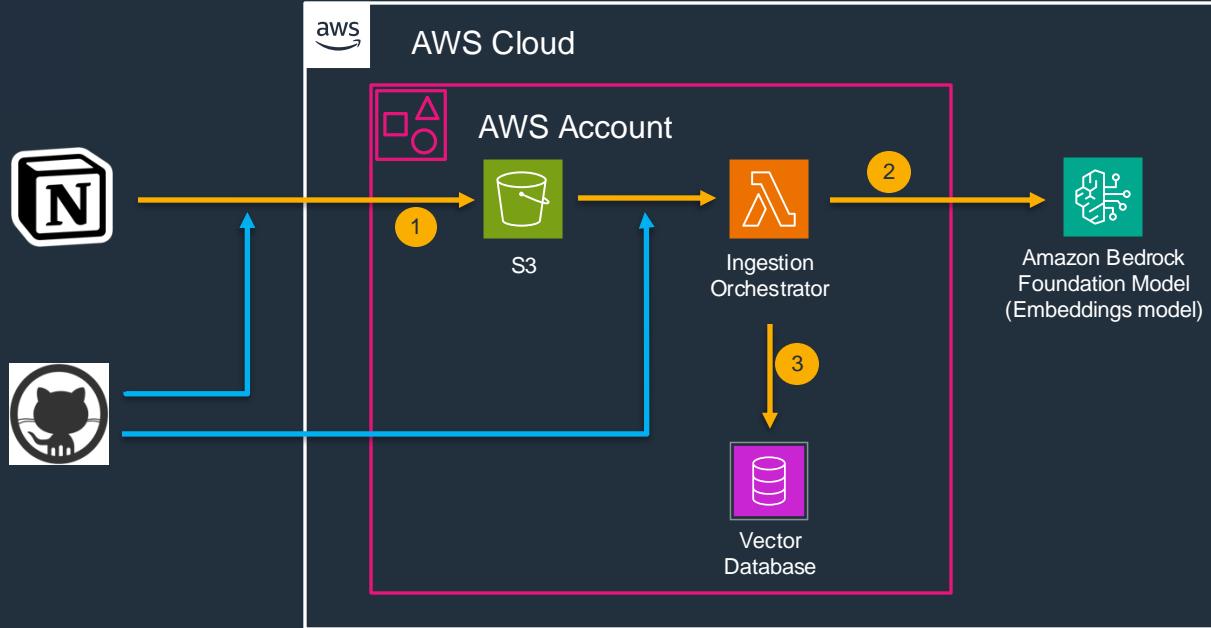
- Fully Managed RAG workflow
- 데이터 소스 통합
 - Amazon S3, Confluence, Salesforce, SharePoint 등
- 다양한 벡터 데이터베이스
 - Amazon OpenSearch Serverless, Pinecone, Redis Enterprise Cloud, Amazon Aurora, MongoDB 등
- API 지원
 - Retrieve API 와 RetrieveAndGenerate API

RAG: 질문과 답변



- 1 사용자 요청 전달
- 2 사용자의 질문을 임베딩으로 변환
- 3 검색 – 고객의 질문과 데이터베이스 간의 유사성 검색
- 4 증강 – 본래 언어 형식의 검색결과와 사용자 질문으로 프롬프팅
- 5 생성 – 생성형 AI 모델이 검색결과를 기반으로 자연어 응답을 생성

RAG: 데이터 수집



- 1 S3에 문서 업로드
- 2 문서를 토큰(token)화, 청크(chunk)화 한 후 임베딩 벡터로 변환
- 3 유사성 검색(Similarity Search)을 위해 임베딩과 메타데이터를 벡터 데이터 베이스에 저장

Claude Prompt 예시

1. “\n\nUser:”
User: 당신은 Karrot Careers라는 회사에서 만든 AI 커리어 코치인 Joe로 활동하게 됩니다. 당신의 목표는 사용자에게 커리어 조언을 제공하는 것입니다.

친절한 고객서비스 톤을 유지해야 합니다.
2. 작업 컨텍스트
다음은 사용자에게 답변할 때 참고할 수 있는 직업안내 문서입니다: <안내>{{DOCUMENT}}</안내>
다음은 대화에 대한 몇 가지 중요한 규칙입니다.
 - 항상 Karrot Careers의 AI인 Joe의 캐릭터를 유지하세요
 - 어떻게 대답해야 할지 모르겠다면, 모른다고 말하세요.다음은 일반적인 대화에서 응답하는 방법의 예입니다:
<예시>
3. 어조 컨텍스트
User: 안녕하세요, 당신은 무슨 일을 하나요?
Joe: 안녕하세요! 제 이름은 Joe이고, Karrot Careers에서 커리어 조언을 제공하고 있습니다.
무엇을 도와드릴까요?
</예시>
4. 처리할 배경 데이터
다음은 질문 전의 대화내역 (사용자와 사용자 간의 대화)입니다. 기록이 없는 경우 비어 있을 수 있습니다:
<내역> {{HISTORY}}</내역>
5. 세부작업 설명 및 규칙
다음은 사용자의 질문입니다:
<질문> {{QUESTION}}</질문>
6. 예시
사용자의 질문에 어떻게 응답하나요?
7. 즉시 처리할 데이터
답변하기 전에 먼저 답변에 대해 생각해 보세요. <응답></응답> 태그에 답변을 입력합니다.
8. 즉각적인 작업 설명 또는 요청
Assistant : [Joe] <응답>
9. 단계별로 생각하기/심호흡하기
10. 출력양식
11. “\n\nAssistant:”

Gurumi Prompt

User: 당신은 AWSKRUG에서 활동하는 AI 비서입니다.

항상 친절하고 전문적인 태도로 질문에 답변해 주세요.

답변을 모르면 모른다고 하세요. 답을 지어내려고 하지 마세요.

참고한 링크가 있다면 링크도 알려주세요.

<context>에 정보가 제공 되면, 해당 정보를 사용하여 답변해 주세요.

<context>[벡터 검색 결과]</context>

<question>태그로 감싸진 질문에 답변을 제공하세요.

<question>[질문]</question>

Assistant:

Source code

<https://github.com/awskrug/lambda-gurumi-ai-bot>



Github Action (Deploy, Serverless framework)

```
✓ Deploy to AWS Lambda 🚀

1 ► Run npx serverless deploy --region us-east-1
24 DOTENV: Loading environment variables from .env:
25   - ALLOWED_CHANNEL_IDS
26   - BOT_CURSOR
27   - DYNAMODB_TABLE_NAME
28   - KNOWLEDGE_BASE_ID
29   - MODEL_ID_TEXT
30   - SLACK_BOT_TOKEN
31   - SLACK_SIGNING_SECRET
32   - PERSONAL_MESSAGE
33   - SYSTEM_MESSAGE
34   - SLACK SAY INTERVAL
35
36 Deploying lambda-gurumi-ai-bot to stage dev (us-east-1)
37 Warning: Function mention has timeout of 600 seconds, however, it's attached to API Gateway so it's automatically limited to 30 seconds.
38
39 ✓ Service deployed to stack lambda-gurumi-ai-bot-dev (59s)
40
41 endpoint: POST - https://████████.execute-api.us-east-1.amazonaws.com/dev/slack/events
42 functions:
43   mention: lambda-gurumi-ai-bot-dev-mention (18 MB)
44
45 1 deprecation found: run 'serverless doctor' for more details
```

Knowledge Bases for Amazon Bedrock

The screenshot shows the AWS Amazon Bedrock console interface. The top navigation bar includes the AWS logo, service menu, search bar, and account information (Barzilina Nalbam). The main navigation bar lists services like Elastic Kubernetes Service, EC2, Lambda, VPC, Route 53, S3, AWS DeepRacer, and Amazon Bedrock.

The left sidebar contains several sections:

- Amazon Bedrock**: A dropdown menu with options like '시작하기' (Getting Started), '개요' (Overview), '예시' (Examples), and '제공업체' (Partners).
- 파운데이션 모델**: Options for foundation models, including '기반 모델' (Model), '사용자 지정 모델' (Custom Model), and 'Imported models' (Preview).
- 플레이그라운드**: Options for playgrounds, including '체팅' (Chat), '텍스트' (Text), and '이미지' (Image).
- okane 스택레이션**: Options for prompt management, including '지식 기반' (Knowledge Base), '에이전트' (Agent), and 'Prompt flows' (Preview).
- 세이프가드**: Options for safety, including '가드레일' (Guardrail) and '워터마크 감지' (Watermark Detection).
- Inference**: Options for inference, including '프로비저닝된 처리량' (Provisioned Processing) and 'Batch inference' (New).

The main content area is titled '지식 기반' (Knowledge Base) and includes tabs for 'Chat with your document' and '지식 기반' (Knowledge Base). It features four main sections:

- 작동 방식** (How it works):
 - Upload and chat**: Quickly query foundation models with context provided by ad-hoc dataset. Includes a 'Chat with your document' button.
 - 지식 기반 생성**: Quickly generate knowledge bases with context provided by ad-hoc dataset.
 - 지식 기반 테스트**: Test the knowledge base with a dataset.
 - 지식 기반 사용**: Use the knowledge base in an application.
- 지식 기반 (1)**: A table listing the generated knowledge base.

이름	상태	설명	소스 파일	생성 시간	마지막 ...	마지막 동기화
gurumi-k...	Ready	-	-	September...	-	-

At the bottom, there are footer links for CloudShell,的意见 (Feedback), and copyright information: © 2024, Amazon Web Services, Inc. 또는 계열사. 개인 정보 보호 약관 쿠키 기본 설정.

New Knowledge Bases

The screenshot shows the AWS Amazon Bedrock console interface. On the left, there's a sidebar with various sections like '시작하기', '파운데이션 모델', '플레이그라운드', '오케스트레이션', '세이프가드', and 'Inference'. The main area is titled 'Amazon Bedrock > 지식 기반 > 지식 기반 생성'. It's a step-by-step wizard:

- 단계 1: 지식 기반 세부 정보 제공**
 - 지식 기반 세부 정보 제공**
 - 데이터 소스 설정
 - 임베딩 모델 선택 및 벡터 저장소 구성
 - 검토 및 생성
- 지식 기반 세부 정보**

지식 기반 이름: (This field is highlighted with a yellow box.)
유효한 문자는 a~z, A~Z, 0~9, _(밑줄) 및 -(하이픈)입니다. 이름은 최대 50자까지 입력할 수 있습니다.

지식 기반 설명 - 선택 사항:
유효한 문자는 a~z, A~Z, 0~9, _(밑줄) 및 -(하이픈)입니다. 이름은 최대 200자까지 입력할 수 있습니다.
- IAM 권한**

IAM 역할은 사용자를 대신하여 다른 서비스에 액세스하는 데 사용됩니다.[service role](#) for Amazon Bedrock

새 서비스 역할 생성 및 사용

기존 서비스 역할 사용
- 서비스 역할 이름**:
- Choose data source**

Select the data source that you want to configure in the next step. You can add up to 5 data sources in a knowledge base.

At the bottom, there are navigation links: CloudShell, 의견, © 2024, Amazon Web Services, Inc. 또는 계열사, 개인 정보 보호, 악관, 쿠키 기본 설정.

Knowledge Bases – Data source

The screenshot shows the AWS Amazon Bedrock console interface. On the left, there's a sidebar with various sections like '시작하기', '파운데이션 모델', '플레이그라운드', '오케스트레이션', '세이프가드', and 'Inference'. The main area is titled 'Choose data source type' and shows three steps: Step 1 (selected), Step 2, and Step 3. Step 1 is titled 'Choose data source' and lists several options: 'Amazon S3' (selected), 'Web Crawler - Preview', 'Confluence - Preview', and 'Salesforce - Preview'. Below these, there's a section for 'Third party data sources' with 'Chromecast - Preview'. At the bottom, there are links for 'CloudShell' and '의견'.

AWS | 서비스 | 검색 [옵션+S] | 버지니아 북부 | sso-nalbam-me/nalbam

Amazon Bedrock < Amazon Bedrock > 지식 기반

Step 1
Choose data source type

Step 2
Configure data source

Step 3
Review and add

Choose data source
Select the data source that you want to configure in the next step

Amazon S3
Object storage service that stores data as objects within buckets.

Web Crawler - Preview
Web page crawler that extracts content from public web pages you are authorized to crawl.

Confluence - Preview
Collaborative work-management tool designed for project planning, software development and product management.

Salesforce - Preview
Customer relationship management (CRM) tool for managing support, sales, and marketing data.

Chromecast - Preview

CloudShell 의견 © 2024, Amazon Web Services, Inc. 또는 계열사. 개인 정보 보호 약관 쿠키 기본 설정

Knowledge Bases – Data source

The screenshot shows the Amazon Bedrock console interface. On the left, there's a sidebar with various navigation options like '시작하기', '파운데이션 모델', '플레이그라운드', '오캐스트레이션', '세이프가드', and 'Inference'. The main content area is titled 'Amazon Bedrock > 지식 기반 > 지식 기반 생성'. A step-by-step guide on the left says '단계 1: 지식 기반 세부 정보 제공', '단계 2: 데이터 소스 설정' (which is selected), '단계 3: 임베딩 모델 선택 및 벡터 저장소 구성', and '단계 4: 검토 및 생성'. The right side is titled '데이터 소스 설정' and 'Amazon S3 정보'. It shows a form for connecting Amazon Bedrock to an S3 data source, with fields for '데이터 소스 이름' (set to 'gurumi-knowledge-base'), 'Data source location' (set to 'This AWS account'), and 'S3 URI' (set to 's3://gurumi-ai-bot-968005369378'). Below this, there's a section for 'Chunking and parsing configurations' with 'Default' and 'Custom' options. At the bottom, there's a note about '고급 설정 - 선택 사항'.

aws | 서비스 | 검색 [옵션+S]

Elastic Kubernetes Service EC2 Lambda VPC Route 53 S3 AWS DeepRacer Certificate Manager

Amazon Bedrock > 지식 기반 > 지식 기반 생성

데이터 소스 설정

Amazon S3 정보

데이터 소스 이름: gurumi-knowledge-base

Data source location: This AWS account

S3 URI: s3://gurumi-ai-bot-968005369378

Chunking and parsing configurations

Default

Custom

고급 설정 - 선택 사항

© 2024, Amazon Web Services, Inc. 또는 계열사. 개인 정보 보호 약관 쿠키 기본 설정

Chunking and parsing configurations

The screenshot shows the AWS Lambda service configuration page for a function named 'HelloWorld'. The 'Handler' dropdown is set to 'lambda.lambda_handler'. The 'Runtime' dropdown is set to 'Python 3.9'. The 'Memory size' input field is set to '128'. The 'Timeout' input field is set to '3'. The 'Layers' section shows a single layer named 'HelloWorldLayer'.

Chunking and parsing configurations

Choose between default or advanced customization.

Default
Uses default parsing and chunking strategy.

Custom
Customize the parsing and chunking strategy, including using advanced parsing.

Parsing strategy

Parsing analyses and extracts useful information from documents.

Use foundation model for parsing [See supported formats](#)

Suitable for parsing more than standard text in supported document formats, including tables within PDFs with their structure intact. [View pricing](#)

Chunking

Chunking은 임베드하기 전에 텍스트를 더 작은 세그먼트로 나눕니다. Chunking은 데이터 소스를 만든 후에는 수정할 수 없습니다.

기본 척킹

기본적으로 지식 기반은 소스 데이터를 각각 최대 300개의 토큰을 포함하는 척크로 자동 분할합니다. 문서에 300개 미만의 토큰이 ...

기본 척킹

기본적으로 지식 기반은 소스 데이터를 각각 최대 300개의 토큰을 포함하는 척크로 자동 분할합니다. 문서에 300개 미만의 토큰이 포함되어 있으면 더 이상 분할되지 않습니다.

고정 크기 척킹

Amazon Bedrock은 텍스트를 거의 일정한 크기의 척크로 분할합니다.

Hierarchical chunking

Organizes text chunks (nodes) into hierarchical structures of parent-child relationships. Each child node includes a reference to its parent node.

Semantic chunking

Organizes text chunks or groups of sentences by how semantically similar they are to each other.

첨킹 없음

이 옵션은 문서를 별도의 파일로 분할하여 사전 처리한 경우, Bedrock이 문서를 더 이상 척크로 만들지 않도록 하려는 경우에 적합합니다.

You can add 4 more data source(s).

취소 이전 다음

CloudShell 의견

© 2024, Amazon Web Services, Inc. 또는 계열사. 개인 정보 보호 약관 쿠키 기본 설정

Chunking and parsing configurations

- Fixed-size chunking (고정 크기 청킹)
 - 데이터를 고정된 크기의 청크로 분할합니다.
 - 일정한 크기의 데이터 블록을 필요로 하는 경우 유용합니다.
- Default chunking (기본 청킹)
 - 데이터를 최대 300개의 토큰으로 분할합니다.
 - 기본적인 텍스트 분할이 필요한 경우에 적합합니다.
- Hierarchical chunking (계층적 청킹)
 - 데이터를 부모-자식 청크의 중첩 구조로 조직합니다.
 - 복잡한 문서 구조를 가진 경우 유용합니다.
- Semantic chunking (의미론적 청킹)
 - 의미적으로 유사한 텍스트 청크나 문장 그룹으로 데이터를 조직합니다.
 - 텍스트의 의미적 유사성을 유지하면서 분할이 필요한 경우 유용합니다.
- No chunking (청킹 없음)
 - 각 문서를 단일 텍스트 청크로 처리합니다.

Knowledge Bases - Embedding Models

The screenshot shows the AWS Amazon Bedrock console interface. On the left, there's a sidebar with various sections like '시작하기', '파운데이션 모델', '플레이그라운드', '오케스트레이션', '세이프가드', and 'Inference'. The main content area is titled 'Amazon Bedrock > 지식 기반 > 지식 기반 생성' and shows a step-by-step process: '단계 1: 지식 기반 세부 정보 제공', '단계 2: 데이터 소스 설정', '단계 3: 임베딩 모델 선택 및 벡터 저장소 구성', and '단계 4: 검토 및 생성'. A large callout box highlights the '임베딩 모델 선택 및 벡터 저장소 구성' step, specifically the '임베딩 모델 선택' section. It lists several models: 'Titan Text Embeddings v2' (selected, from Amazon), 'Titan Embeddings G1 - Text v1.2' (from Amazon), 'Embed English v3' (from Cohere), and 'Embed Multilingual v3' (from Cohere). Below this, there's a section for '벡터 치수' (Vector Dimension) with a dropdown set to '1024'. The top navigation bar includes links for AWS services like Elastic Kubernetes Service, EC2, Lambda, VPC, Route 53, S3, AWS DeepRacer, Certificate Manager, and the current page 'Amazon Bedrock'. The bottom of the screen shows standard AWS footer links for CloudShell,意见反馈 (Feedback), Copyright 2024, and various legal notices.

Knowledge Bases - Vector Database

The screenshot shows the AWS Amazon Bedrock console interface. On the left sidebar, there are several sections: **시작하기** (Getting Started) with links to '개요', '예시', and '제공업체'; **파운데이션 모델** (Foundation Models) with '기반 모델' and '사용자 지정 모델'; **Imported models** (Preview); **플레이그라운드** (Playground) with '체팅', '텍스트', and '이미지'; **오키스트레이션** (Orchestrator) with 'Prompt management' (Preview), '지식 기반', '에이전트', and 'Prompt flows' (Preview); **세이프가드** (SafeGuard) with '가드레일' and '워터마크 감지'; and **Inference** with '프로비저닝된 처리량' and 'Batch inference' (New).

The main content area displays a card for the **Embed Multilingual v3** model by **기준 Cohere**. It includes a dropdown for **벡터 차수** (Vector Dimension) set to 1024, with a note that higher dimensions improve accuracy but require more storage. Below this, there are two options for **벡터 데이터베이스** (Vector Database): **새로운 벡터 저장소 빠른 생성 - 권장** (Quick Create New Vector Storage - Recommended) and **생성한 벡터 저장소를 선택하세요** (Select Existing Vector Storage). The first option is highlighted with a yellow border. At the bottom, there are checkboxes for **이중화 활성화 (활성 복제본) - 선택 사항** (Enable Duplication (Active Replicas) - Optional) and **Amazon OpenSearch Serverless 벡터를 위한 고객 관리형 KMS 키 추가 - 선택 사항** (Add Customer-managed KMS Key for Amazon OpenSearch Serverless Vector - Optional). Navigation buttons at the bottom right include '취소' (Cancel), '이전' (Previous), and '다음' (Next).

Page footer: © 2024, Amazon Web Services, Inc. 또는 계열사. 개인 정보 보호 약관 쿠키 기본 설정

Knowledge Bases - Vector Database

The screenshot shows the AWS Amazon Bedrock console interface. On the left, a sidebar lists various features: 시작하기 (Getting Started), 파운데이션 모델 (Foundation Model), 플레이그라운드 (Playground), 오케스트레이션 (Orchestration), 세이프가드 (SafeGuard), and Inference. The '시작하기' section is expanded, showing '개요' (Overview), '예시' (Examples), '제공업체' (Partners), '파운데이션 모델' (Foundation Model) which includes '기반 모델' (Base Model), '사용자 지정 모델' (Custom Model), and 'Imported models' (Preview); '플레이그라운드' (Playground) which includes '체팅' (Chat), '텍스트' (Text), and '이미지' (Image); '오케스트레이션' (Orchestration) which includes 'Prompt management' (Preview) and 'Prompt flows' (Preview); '세이프가드' (SafeGuard) which includes '가드레일' (Guardrail) and '워터마크 감지' (Watermark Detection); and 'Inference' which includes '프로비저닝된 처리량' (Provisioned Processing) and 'Batch inference' (New).

The main content area shows a progress bar for creating a knowledge base:

- 단계 1: 지식 기반 세부 정보 제공 (Step 1: Knowledge base detailed information provision)
- 단계 2: 데이터 소스 설정 (Step 2: Data source configuration)
- 단계 3: 임베딩 모델 선택 및 벡터 저장소 구성 (Step 3: Embedding model selection and vector storage configuration)
- 단계 4: 검토 및 생성 (Step 4: Review and Create) - This step is selected.

검토 및 생성 (Review and Create) screen details:

- 1단계: 세부 정보 제공** (Step 1: Detailed information provision):
 - 지식 기반 세부 정보** (Knowledge base detailed information):
 - 지식 기반 이름: gurumi-knowledge-base
 - 지식 기반 설명: —
 - 서비스 역할: AmazonBedrockExecutionRoleForKnowledgeBase_2sgbm
 - Data source type** (Data source type): S3
- 태그 (0)** (Tags (0)): 표시할 태그 없음 (No tags to display).

2단계: 데이터 소스 설정 (Step 2: Data source configuration):
데이터 소스: gurumi-knowledge-base

Page footer: CloudShell, 의견, © 2024, Amazon Web Services, Inc. 또는 계열사, 개인 정보 보호, 약관, 쿠키 기본 설정

Knowledge Bases

The screenshot shows the AWS Amazon Bedrock Knowledge Base management interface. The left sidebar contains navigation links for various services like Elastic Kubernetes Service, EC2, Lambda, VPC, Route 53, S3, AWS DeepRacer, and Amazon Bedrock. The main content area displays a knowledge base named "gurumi-knowledge-base".

Amazon Bedrock > 지식 기반 > gurumi-knowledge-base

gurumi-knowledge-base

지식 기반 개요

지식 기반 이름: gurumi-knowledge-base

지식 기반 ID: GMO2VR1WVQ

상태: Ready

생성 날짜: September 19, 2024, 12:15 (UTC+09:00)

Log Deliveries
Configure log deliveries and event logs in the [Edit](#) page.

태그

태그는 AWS 리소스에 할당하는 레이블입니다. 각 태그는 키와 선택적 값으로 구성됩니다. 태그를 사용하여 리소스를 검색 및 필터링하거나 AWS 비용을 추적할 수 있습니다.

Key	Value
태그 없음	

[태그 관리](#)

데이터 소스 (1)

[추가](#) [편집](#) [삭제](#) [동기화](#)

지식 기반 테스트

응답 생성

Claude 3.5 Sonnet v | On-demand throughput
1

변경

답레이서 타이머가 뭐야?

답레이서 타이머는 AWS DeepRacer 경주에서 랙타임을 측정하고 기록하는 도구입니다. 이 타이머는 압력 감지 센서를 사용하여 DeepRacer 차량이 출발/도착선을 통과할 때를 감지하고, 라즈베리파이와 연결되어 시간을 측정합니다. 타이머 시스템은 웹 인터페이스를 통해 제어할 수 있으며, 시작, 일시정지, 통과, 리셋, 클리어 등의 기능을 제공합니다. 이 시스템은 DeepRacer 대회에서 정확한 랙 타임을 측정하고 리더보드를 관리하는 데 사용됩니다.[\[자세히\]](#) 이 타이머 시스템은 GitHub에 공개되어 있어 누구나 사용하고 개선할 수 있습니다. 또한, 온라인에서 선서 없이 테스트 해볼 수 있는 버전도 제공됩니다. 이 프로젝트는 DeepRacer 커뮤니티에서 영감을 받아 개발되었으며 실제 대회에서 사용되어 그 유용

여기에 메시지를 입력하세요. [▶ 실행](#)

CloudShell 의견 © 2024, Amazon Web Services, Inc. 또는 계열사. 개인 정보 보호 약관 쿠키 기본 설정

Amazon OpenSearch Serverless

The screenshot shows the AWS Amazon OpenSearch Service console. The left sidebar has a tree view with the following expanded sections:

- 관리형 클러스터**: 대시보드, 도메인, 예약 인스턴스 리스트, 페키지, VPC 엔드포인트.
- 서비스**: 대시보드, **컬렉션 신규** (highlighted with a yellow box).
- 수집**: Dashboard, 파이프라인.

The main content area shows the details for the collection "bedrock-knowledge-base-87t9d1". The top navigation bar includes the service name, region (Asia Pacific (Seoul)), and account ID (sso-nalbam-me/nalbam). The breadcrumb path is "Amazon OpenSearch Service > 서비스: 컬렉션 > bedrock-knowledge-base-87t9d1". A note says "벡터 데이터 인덱싱을 시작하세요. 벡터 인덱스는 데이터를 설명하는 메타데이터와 함께 벡터 임베딩으로 구성됩니다." (Start indexing vector data. Vector indexes store data along with meta-data and vector embeddings). Buttons for "Delete collection" and "OpenSearch 대시보드" are present.

bedrock-knowledge-base-87t9d1 정보

개요	모니터링	인덱스	태그
일반 정보	설명 편집		
상태 활성	수집 설명 Default collection created by Amazon Bedrock Knowledge base.	생성 날짜 9월 19, 2024, 12:02 (UTC+09:00)	
수집 유형 Vectorsearch	총 크기 193.39 KiB	수집 ARN arn:aws:aoss:us-east-1:968005369378:collection/p1r1svz7trl7sfvqxlm1	
인덱스 1	배포 유형 개발 및 테스트 모드가 활성화됨		

엔드포인트

OpenSearch 엔드포인트	OpenSearch 대시보드 URL
https://p1r1svz7trl7sfvqxlm1.us-east-1.aoess.amazonaws.com	https://dashboards.us-east-1.aoess.amazonaws.com/_login/

CloudShell 의견 © 2024, Amazon Web Services, Inc. 또는 계열사. 개인 정보 보호 약관 쿠키 기본 설정

Notion export – 날밤 스페이스

nalbam

날밤 연구소 / 팀스페이스 홈 Restricted

Share ⌂ ⓘ ☆ ⋮

Search

Notion AI

Home

Inbox

Teamspace

날밤 연구소

팀스페이스 홈

Getting Started

Calendar

Templates

Trash

Help & support

Invite members

날밤 연구소 / 팀스페이스 홈 Restricted

팀스페이스 홈

다른 동료들이 우리 팀과 팀의 업무를 알아갈 수 있는 공간을 마련해 주세요. 왼쪽 사이드바의 + 버튼을 눌러 페이지를 추가할 수 있습니다. 프로세스 문서나 프로젝트 로드맵을 추가해보는 건 어떨까요?

👋 우리 팀에 오신 걸 환영합니다! 우리 팀의 미션은요...

질문이 있으신가요? 아래 연락처로 알려주세요.

- Slack 채널: <https://nalbam.slack.com>
- 이메일: me@nalbam.com
- 팀장: @Jungyoul Yu

팀 소개

@를 입력하고 팀원을 태그하세요

딥레이서 타이머 및 센서 제작기

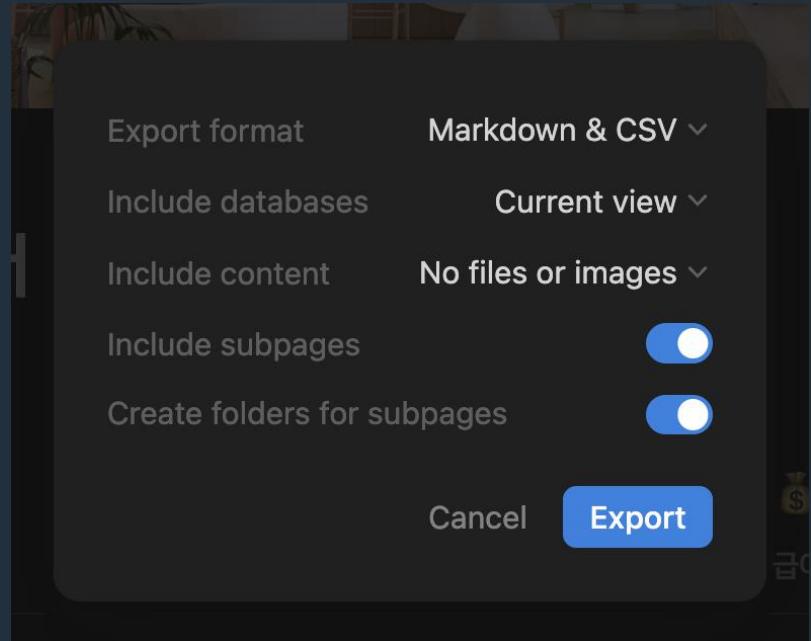
AWS 클라우드, 라즈베리파이, 온도센서를 활용한 체온 알람 서비스

팀 리소스

/를 입력하고 원하는 블록을 추가하세요

Notion export

- Markdown 파일
- 파일, 이미지 제외 가능
- 하위 문서 가능
- 링크 문서 안됨
- 중복 가능성
- 출처 링크 이슈



Github Action - Notion export and Sync

The screenshot shows a GitHub Actions workflow configuration in a code editor. The left sidebar displays the repository structure, including files like `5-sync-notion.yml`, `6-start-ingestion.yml`, and `push.yml`. The main editor area shows the `5-sync-notion.yml` file with syntax highlighting for YAML. A yellow box highlights the following configuration block:

```
NOTION_PAGE_NAME: "nalbam"
NOTION_PAGE_ID: "7aac0412a82431996f61a29225a95ec"
```

The file also contains other configurations such as environment variables, schedules, and job definitions.

Github Action - Notion export and Sync

sync
succeeded 11 minutes ago in 22s

Search logs

> Set up job 1s

> Checkout 1s

> Setup Python 3.11 0s

> Setup Dependencies 5s

> Run Notion Exporter 7s

```
1 ▶ Run python bin/notion_exporter.py
20
21  0%|          | 0/1 [00:06<-?, ?it/s]
22  0%|          | 0/1 [00:06<?, ?it/s, Exporting nalbam... 3 pages already exported]
23 100%|[██████| 1/1 [00:06<0:00,  6.60s/it, Exporting nalbam... 3 pages already exported]
24 100%|[██████| 1/1 [00:06<0:00,  6.60s/it, Exporting nalbam... 3 pages already exported]
```

> Sync to AWS S3 Data Source 7s

```
1 ▶ Run aws s3 sync --delete --region us-east-1 \
2 Completed 5.7 KiB/11.8 KiB (117.1 KiB/s) with 3 file(s) remaining
3 upload: build/nalbam/날밤이서 타이머 및 센서 제작기 e518e48adaa7464b832db28b47d89bc2.md to s3://gurumi-ai-bot-968005369378/nalbam/날밤이서 타이머 및 센서 제작기 e518e48adaa7464b832db28b47d89bc2.md
4 Completed 5.7 KiB/11.8 KiB (117.1 KiB/s) with 2 file(s) remaining
5 Completed 10.2 KiB/11.8 KiB (108.5 KiB/s) with 2 file(s) remaining
6 upload: build/nalbam/AWS 클라우드, 라즈베리파이, 온도센서를 활용해 fac01fe356924134a8fbf1ab21baf16c.md to s3://gurumi-ai-bot-968005369378/nalbam/AWS 클라우드, 라즈베리파이, 온도센서를 활용해 fac01fe356924134a8fbf1ab21baf16c.md
7 Completed 10.2 KiB/11.8 KiB (108.5 KiB/s) with 1 file(s) remaining
8 Completed 11.8 KiB/11.8 KiB (122.5 KiB/s) with 1 file(s) remaining
9 upload: build/nalbam/팀스페이스 훔 7aace0412a82431996f61a29225a95ec.md to s3://gurumi-ai-bot-968005369378/nalbam/팀스페이스 훔 7aace0412a82431996f61a29225a95ec.md
```

> Sync to AWS Bedrock Knowledge Base 1s

> Post Setup Python 3.11 0s

> Post Checkout 0s

> Complete job 0s

Data source - Amazon S3

The screenshot shows the AWS S3 console interface. At the top, the AWS logo and navigation links like '서비스' (Services), '검색' (Search), '[옵션+S]', and '버지니아 북부' (Virginia North) are visible. Below the navigation bar, the breadcrumb trail indicates the path: 'Amazon S3 > 버킷 > gurumi-ai-bot-968005369378 > nalbam/'. On the right side, there are several icons including 'S3 URI 복사' (Copy S3 URI). The main area displays the contents of the 'nalbam/' bucket. The '객체' (Objects) tab is selected, showing three objects:

이름	유형	마지막 수정	크기
딥레이저 타이머 및 센서 제작기 e518e48adaa7464b832db28b47d89bc2.md	md	2024. 9. 19. am 5:04:25 AM KST	5.7KB
팀스페이스 홈 7aaice0412a82431996f61a29225a95ec.md	md	2024. 9. 19. am 5:04:25 AM KST	1.6KB
AWS 클라우드, 라즈베리파이, 운도센서를 활용한 fac01fe356924134a8fbfb1ab21baf16c.md	md	2024. 9. 19. am 5:04:25 AM KST	4.6KB

At the bottom, there are links for 'CloudShell', '의견' (Feedback), '© 2024, Amazon Web Services, Inc. 또는 계열사.', '개인 정보 보호' (Privacy), '약관' (Terms of Service), and '쿠키 기본 설정' (Cookie Preferences).

Data source - Sync

The screenshot shows the AWS Bedrock interface for managing data sources and embeddings.

Left Sidebar:

- Elastic Kubernetes Service
- EC2
- Lambda
- VPC
- Route 53
- S3
- AWS DeepRacer
- Amazon Bedrock

Top Bar:

- aws
- 서비스
- 검색
- [옵션+S]
- 버지니아 북부
- sso-nalbam-me/nalbam

Main Content Area:

데이터 소스 (1)

메타데이터를 .metadata.json 파일로 S3 버킷에 추가하여 응답의 정확성과 관련성을 높이세요.

데이터 소스	상태	Account ID	Source Link	마지막 통기일	마지막 통지일	첨경 전략	
gurumi-k...	Available	S3	96800536...	s3://guru...	September...	-	Semantic chunking.....

Buttons: 추가, 편집, 삭제, 동기화 (highlighted with a yellow box).

임베딩 모델

모델: Titan Text Embeddings v2 | Vector dimensions: 1024

벡터 데이터베이스

벡터 데이터베이스: 벡터 엔진 Amazon OpenSearch Serverless

컬렉션 ARN: arn:aws:aoss:us-east-1:968005369378:collection/p1r1svz7rl7sfvqlm1	벡터 필드: bedrock-knowledge-base-default-vector
벡터 인덱스 이름: bedrock-knowledge-base-default-index	Bedrock 관리형 메타데이터 필드: AMAZON_BEDROCK_METADATA
텍스트 필드: AMAZON_BEDROCK_TEXT_CHUNK	

Bottom Navigation:

- CloudShell
- 의견
- © 2024, Amazon Web Services, Inc. 또는 계열사.
- 개인 정보 보호
- 약관
- 쿠키 기본 설정

Bedrock Agent Retrieve

```
nalbam@Karrot-Bruce:~/workspace/github.com/nalbam/lambda-gurumi-ai-bot/bedrock
(*)|orbstack:default)→ bedrock (main) ✘ python invoke_knowledge_base.py -p "딥 레이서 타이머 ?"
invoke_knowledge_base: # 딥레이서 타이머 및 센서 제작기
원본
- [Korean](https://nalbam.github.io/2019/11/07/deepracer-timer-ko.html), [English](https://nalbam.github.io/2019/11/07/deepracer-timer-en.html)
저는 지난 해 말 AWS re:Invent 2018에 참가했습니다. AWS는 딥레이서를 발표하였고, 워크샵에 참가하여 딥레이서를 받았습니다. 그때부터 딥레이서를 훈련하였고 몇몇 대회에 참가하고 나쁘지 않은 성적도 거두었습니다. 서울 서밋에서 9등, 도쿄 서밋에서 22등... 그리고 온라인 리그 6개 시즌 포인트 합산 순위 6등으로 AWS re:Invent 2019에서 열리는 결승전에 참가 할 수 있게 되었습니다.
![https://nalbam.github.io/assets/images/2019-11-07/learderboard.png](https://nalbam.github.io/assets/images/2019-11-07/learderboard.png)
딥레이서 커뮤니티에서 많은 정보를 얻었는데, 그 중 어떤 사용자가 딥레이서 타이머를 만들면서 레이저 센서로 랙타임을 체크하는 사진을 보게 되었습니다. 하지만 레이저 센서 타워(?)를 트랙에 세우게 되면 딥레이서와 충돌 할 수 있겠다는 생각이 들었고, 이것을 해결하고 나도 타이머를 만들고 싶다는 욕구가 생겼습니다.
그러던 중 메가존 클라우드로 이직하게 되었고, 메가존 클라우드가 후원하는 딥레이서 대회에서 PitBoss로 대회를 지원하는 역할을 했습니다. 딥레이서 타이머와 리더보드를 만들었고, [압력 감지 센서 (SEN0299)](https://www.eleparts.co.kr/goods/view?no=7643152)를 라즈베리파이에 GPIO로 연결해 보았습니다.
!

[https://nalbam.github.io/assets/images/2019-11-07/rpi2.jpg](https://nalbam.github.io/assets/images/2019-11-07/rpi2.jpg)
몇년 만에 납땜질
변환기는 4개의 핀이 있는데, VCC, GND, AOUT, DOUT입니다. VCC는 라즈베리파이의 3V에 연결해 주었고, GND는 GND에 연결했습니다. AOUT는 사용하지 않았고, DOUT은 각각 라즈베리파이의 11번과 13번 핀에 연결해 주었습니다. 그리고 라즈베리파이에 SSH로 접속해서 딥레이서 타이머 소스를 clone하고, 시작 쉘을 실행해 줍니다.
```jsx
```

# Slack - Event Subscriptions

The screenshot shows the Slack API Event Subscriptions settings page for the 'Gurumi' app. The left sidebar lists various settings and features, with 'Event Subscriptions' currently selected. The main content area displays the 'Enable Events' section, which is turned 'On'. It also shows the 'Request URL Verified' field containing a placeholder URL and a 'Change' button. Below this, there are three sections: 'Subscribe to bot events', 'Subscribe to events on behalf of users', and 'App unfurl domains', each with a right-pointing arrow.

slack api

Gurumi

Search Documentation Tutorials Your Apps

Settings

- Basic Information
- Collaborators
- Socket Mode
- Install App
- Manage Distribution

Features

- App Home
- Agents & Assistants NEW
- Workflow Steps NEW
- Org Level Apps
- Incoming Webhooks
- Interactivity & Shortcuts
- Slash Commands
- Steps from Apps LEGACY
- OAuth & Permissions
- Event Subscriptions**
- User ID Translation
- App Manifest
- Beta Features

Event Subscriptions

Enable Events **On**

Your app can subscribe to be notified of events in Slack (for example, when a user adds a reaction or creates a file) at a URL you choose. [Learn more](#).

Request URL **Verified** ✓

https://[REDACTED].execute-api.us-east-1.amazonaws.com/dev/slack/events [Change](#)

We'll send HTTP POST requests to this URL when events occur. As soon as you enter a URL, we'll send a request with a `challenge` parameter, and your endpoint must respond with the challenge value. [Learn more](#).

Subscribe to bot events

Subscribe to events on behalf of users

App unfurl domains

[Discard Changes](#) [Save Changes](#)

# Gurumi (구루미)



**nalbam** 바로 지금

@Gurumi (구루미) 딥레이서 타이머가 뭐야



**Gurumi (구루미)** 앱 바로 지금

딥레이서 타이머는 AWS DeepRacer 자율주행 모델 카의 주행 시간을 측정하는 장치입니다. 이 글에서 설명하는 딥레이서 타이머의 주요 특징은 다음과 같습니다:

1. 라즈베리파이를 사용하여 제작되었습니다.
2. 압력 감지 센서(SEN0299)를 사용하여 딥레이서의 통과를 감지합니다.
3. 웹 인터페이스를 통해 제어할 수 있으며, 시작, 일시정지, 통과, 리셋, 지우기 등의 기능이 있습니다.
4. 키보드로도 조작이 가능합니다 (Q, W, E, R, T 키 사용).
5. 랙타임을 측정하고 기록할 수 있습니다.
6. 리더보드 기능을 포함하고 있어 여러 참가자의 기록을 비교할 수 있습니다.

이 타이머는 AWS DeepRacer 대회나 훈련 세션에서 정확한 주행 시간을 측정하고 기록하는 데 사용될 수 있습니다.

자세한 정보와 소스 코드는 다음 GitHub 저장소에서 확인할 수 있습니다:

<https://github.com/nalbam/deepracer-timer> (편집됨)

# Source code

<https://github.com/awskrug/lambda-gurumi-ai-bot>



A close-up photograph of a wooden crate overflowing with fresh, orange carrots. The carrots are of various sizes and have their green leafy tops still attached. The crate is placed on a surface, and in the background, there are some purple flowers and other out-of-focus produce.

감사합니다.