

# 대화형 쿠버네티스: LLM 기반 K8s 운영 기술의 진화와 가능성

임지훈

# CONTENTS

---

**1 K8sGPT**

---

**2 K8sGPT with Amazon Bedrock**

---

**3 K8s MCP**

---

**4 kubectl-ai**

---

**5 정리**

## 광고

Amazon Bedrock으로 시작하는 실전 생성형 AI 개발



판매 링크 : [예스24](#), [교보문고](#), [알라딘](#)

### Amazon Bedrock을 기초부터 실전까지

생성형 AI의 개념, AWS 가입부터 활용까지 다루고 있어 AI 초심자들도 부담 없이 시작할 수 있습니다.

### 실전에서 바로 적용할 수 있는 AI 개발 과정

프롬프트 엔지니어링의 사례, RAG와 Agent 구현 방법, 모델 학습 등 깊이 있는 방법까지 전달합니다.

### Amazon Bedrock의 최신 기능까지

지식 기반과 프롬프트 관리/흐름, 스튜디오 등 최신 기능들까지 놓치지 않고 두루 섭렵했습니다.

# K8SGPT

K8sGPT 이론 공부

K8sGPT 이론 공부 시간

# K8sGPT 장표로 대체

# K8sGPT Prompt

K8sGPT 프롬프트

## k8sgpt/pkg/ai/prompts.go

### 1. default\_prompt

- Kubernetes 오류 메시지를 단순화하고 280자 이내의 단계별 해결책을 제공
- 출력 형식: "Error: {설명}" + "Solution: {해결책}"

### 2. prom\_conf\_prompt

- Prometheus 구성 파일 검증 시 발생하는 오류 메시지를 단순화
- 300자 이내로 Prometheus 문서를 참조하여 단계별 해결책을 제공

### 3. prom\_relabel\_prompt

- Prometheus의 relabel\_configs와 kubernetes\_sd\_configs를 분석하여, 각 job\_name에 대한 Kubernetes 서비스 및 파드 레이블, 네임스페이스, 포트, 컨테이너 정보를 설명

### 4. kyverno\_prompt

- Kyverno 경고 메시지를 단순화하고 kubectl 명령어 형태의 해결책을 제공

### 5. raw\_promt

- 언어, 메시지, 프롬프트를 포함하는 JSON 형식의 기본 템플릿

### 6. PromptMap

- 각 상황에 맞는 프롬프트 템플릿을 선택할 수 있도록 매핑한 맵

# K8sGPT with Amazon Bedrock

# Amazon Bedrock in K8sGPT

코드로 알아보는 – K8sGPT

## k8sgpt/pkg/ai/bedrock\_support/completions.go

```
92  func isModelSupported(modelName string) bool {
93      for _, supportedModel := range SUPPORTED_BEDROCK_MODELS {
94          if modelName == supportedModel {
95              return true
96          }
97      }
98      return false
99  }
100
101 func (a *AmazonCompletion) GetCompletion(ctx context.Context, prompt
102     if !isModelSupported(modelConfig.ModelName) {
103         return nil, fmt.Errorf("model %s is not supported", m
104     }
```

```
10  var SUPPORTED_BEDROCK_MODELS = []string{
11      "anthropic.claude-3-5-sonnet-20240620-v1:0",
12      "us.anthropic.claude-3-5-sonnet-20241022-v2:0",
13      "anthropic.claude-v2",
14      "anthropic.claude-v1",
15      "anthropic.claude-instant-v1",
16      "ai21.j2-ultra-v1",
17      "ai21.j2-jumbo-instruct",
18      "amazon.titan-text-express-v1",
19      "amazon.nova-pro-v1:0",
20      "eu.amazon.nova-lite-v1:0",
21  }
```

# Amazon Bedrock in K8sGPT

K8sGPT에서 Bedrock 사용하기

## 지원 모델 버전 업데이트 및 Access 부여

**v0.4.2** Latest

 github-actions released this 5 days ago

**0.4.2 (2025-03-28)**

**Features**

- old sonnet ([#1408](#)) ([e5817f9](#))

```
[→ k8sgpt version
k8sgpt: 0.4.2 (Homebrew), built at: 2025-03-28T14:17:21Z]
```

▼ Anthropic (10)	6/10 access granted
Claude 3.7 Sonnet	<a href="#">Cross-region inference</a>
Claude 3.5 Haiku	 액세스 권한 부여됨
Claude 3.5 Sonnet v2	 액세스 권한 부여됨
Claude 3.5 Sonnet	 액세스 권한 부여됨

# Amazon Bedrock in K8sGPT

K8sGPT에서 Bedrock 사용하기

## Bedrock Backend 추가

```
→ k8sgpt auth add -b amazonbedrock -m "anthropic.claude-3-5-sonnet-20241022-v2:0" -r us-west-2  
amazonbedrock added to the AI backend provider list
```

```
→ k8sgpt analyze --explain --interactive --backend amazonbedrock  
Error: model not found
```

```
→ k8sgpt auth add -b amazonbedrock -m "us.anthropic.claude-3-5-sonnet-20241022-v2:0" -r us-west-2  
amazonbedrock added to the AI backend provider list
```

```
→ k8sgpt analyze --explain --interactive --backend amazonbedrock  
0% |  
| (0/13, 0 it/hr) [0s:0s]  
Error: failed while calling AI provider amazonbedrock: ValidationException: "claude-3-5-sonnet-2024  
1022" is not supported on this API. Please use the Messages API instead.
```

# Amazon Bedrock in K8sGPT

CloudWatch로 Model Invocations 확인하기

## Anonymize 옵션 여부

미적용

```
"messages": [
    {
        "content": "Simplify the following Kubernetes error message delimited by triple dashes written in --- english
app.kubernetes.io/component=webhook Service has no endpoints, expected label app.kubernetes.io/instance=cert-manager Service has no
most possible solution in a step by step style in no more than 280 characters. Write the output in the following format:\n\tError:
        "role": "user"
    }
]
```

적용

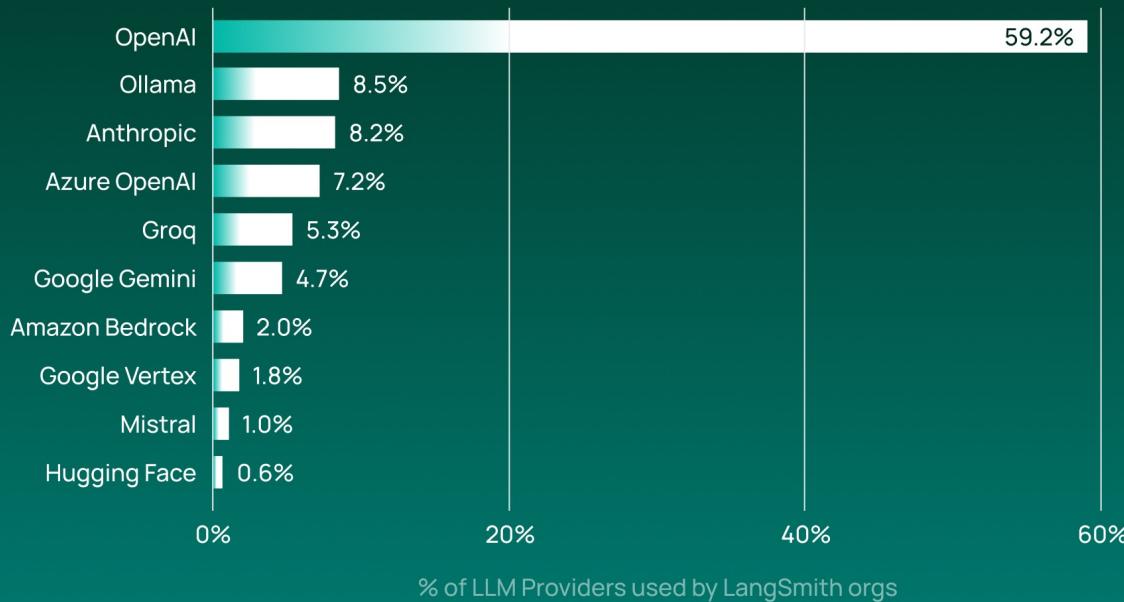
```
"messages": [
    {
        "content": "Simplify the following Kubernetes error message delimited by triple dashes written in --- Korean -
ZULqQFhTQlZYVEVRMD4+WnNqcCRuUWYnJ3FG=X0t4Q3lH Service has no endpoints, expected label MV4qRiNseG55czNoUmVMN0hXJT1hbj1QRUU==KFpBZkt
MjVbYy0zOCZVR0MzYmppL2dq0ChZJQ==Sy5TeGond15T0Q== ---.\n\tProvide the most possible solution in a step by step style in no more tha
error here}\n\tSolution: {Step by step solution here}",
        "role": "user"
    }
]
```

# Amazon Bedrock Access Gateway

Amazon Bedrock을 위한 OpenAI 호환 RESTful API

## OpenAI Provider

### Top 10 LLM Providers 2024



## 주요 기능

### 지원 기능

- server-sent events (SSE)를 통한 스트리밍 응답 지원
- Model APIs
- Chat Completion APIs
- Tool Call
- Embedding API
- Multimodal API
- Cross-Region Inference
- Reasoning

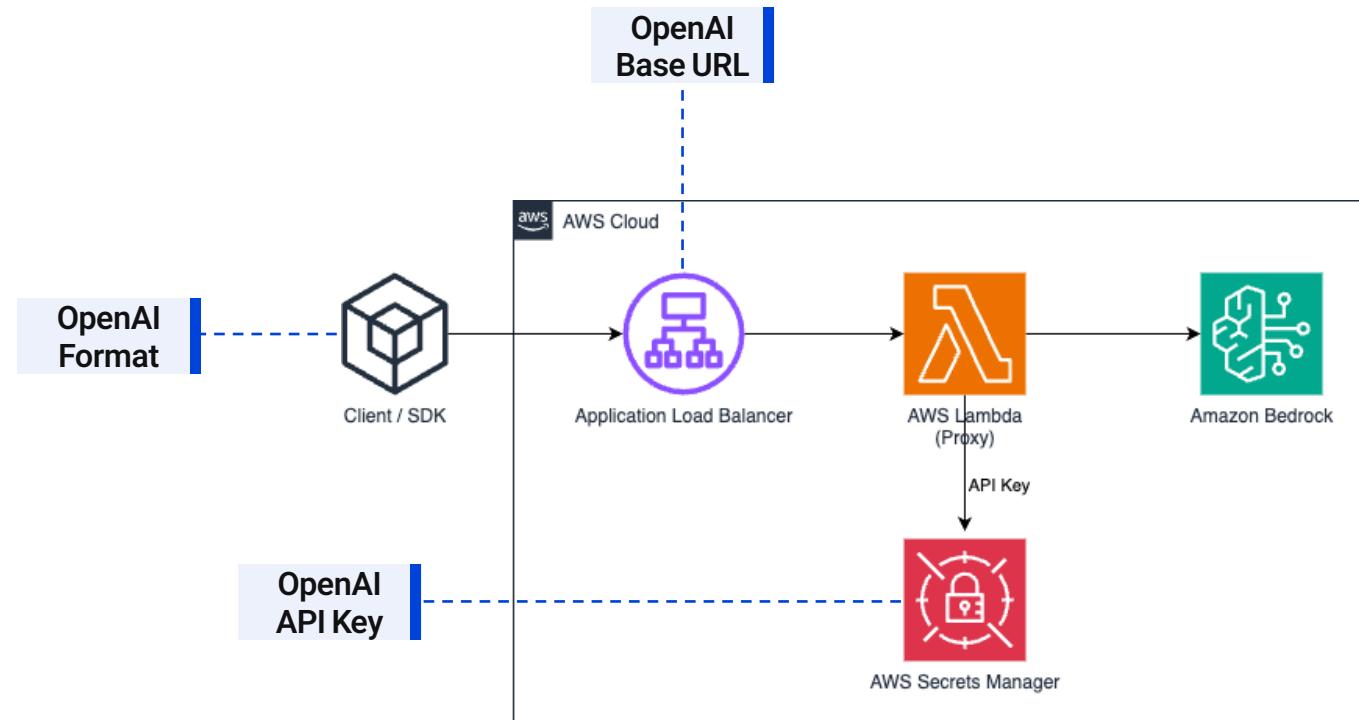
### 미지원 기능

- 구조화된 출력
- Audio 이해
- 이미지 생성

# Amazon Bedrock Access Gateway

Amazon Bedrock을 위한 OpenAI 호환 RESTful API

## OpenAI Provider



## 사용방법

```
export OPENAI_API_KEY=<API key>
export OPENAI_BASE_URL=<API base url>
```

```
from openai import OpenAI
client = OpenAI()

messages = [
    "role": "user", "content": "which one is bigger, 3.9 or 3.11?"
]

response = client.chat.completions.create(
    model="us.anthropic.claude-3-7-sonnet-20250219-v1:0",
    messages=messages,
    reasoning_effort="low",
    max_completion_tokens=4096,
)

reasoning_content =
response.choices[0].message.reasoning_content
content = response.choices[0].message.content
```

# Amazon Bedrock Access Gateway

Amazon Bedrock을 위한 OpenAI 호환 RESTful API

## Example

Model Specification `anthropic.claude-3-sonnet-20240229-v1:0`

Enter parameters for your model.

anthropic.claude-3-sonnet-20240229-v1:0

.....

http://Be 33.us-west-2.elb.amazonaws.com/api/v1

API Type (e.g. azure)

API Version (optional)

Claude 3 Sonnet

ⓘ Model tested successfully

```
$ k8sgpt auth add --backend localai \
--model us.anthropic.claude-3-7-sonnet-20250219-v1:0 \
--password $OPENAI_API_KEY --baseurl $OPENAI_BASE_URL

$ k8sgpt auth default -p localai

$ k8sgpt analyze --explain
```

**K8sGPT**

K8sGPT Demo

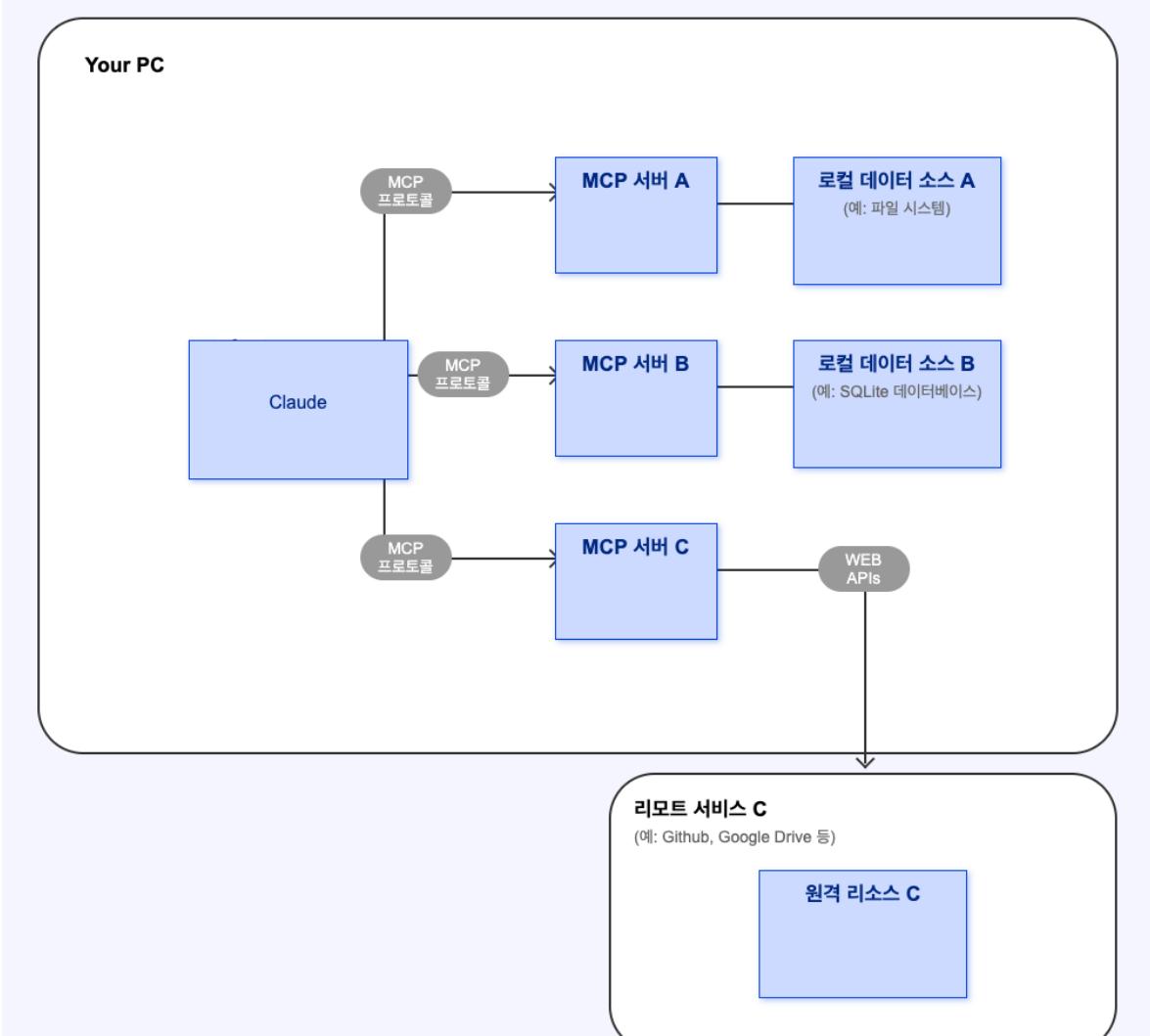
---

# Demo

# K8S MCP

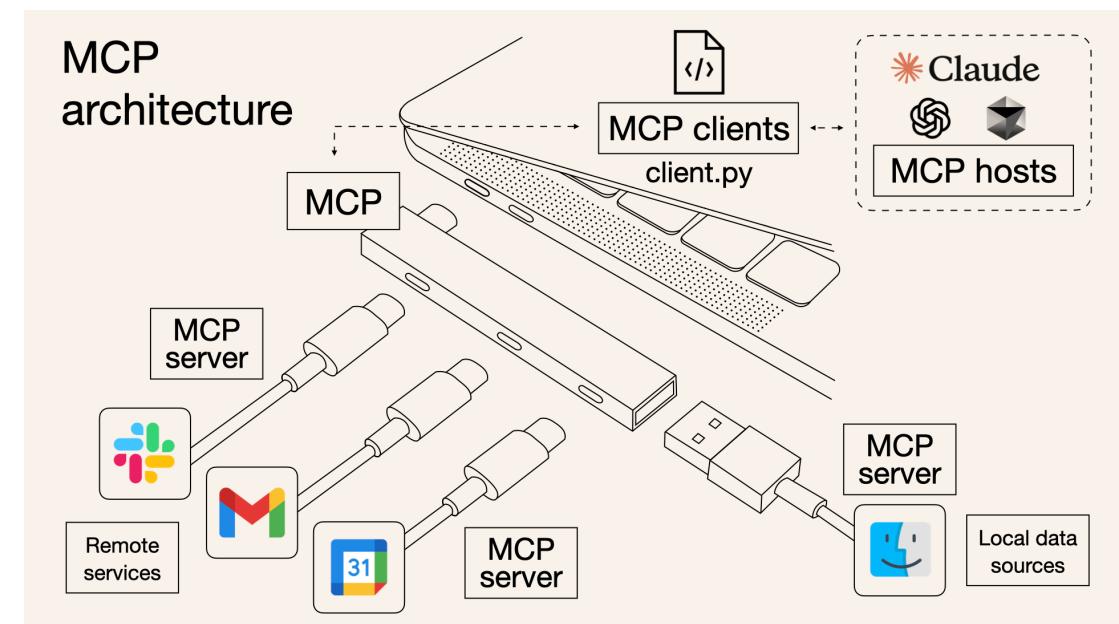
# MCP (Model Context Protocol)

MCP의 기본 개념에 대해 설명합니다.



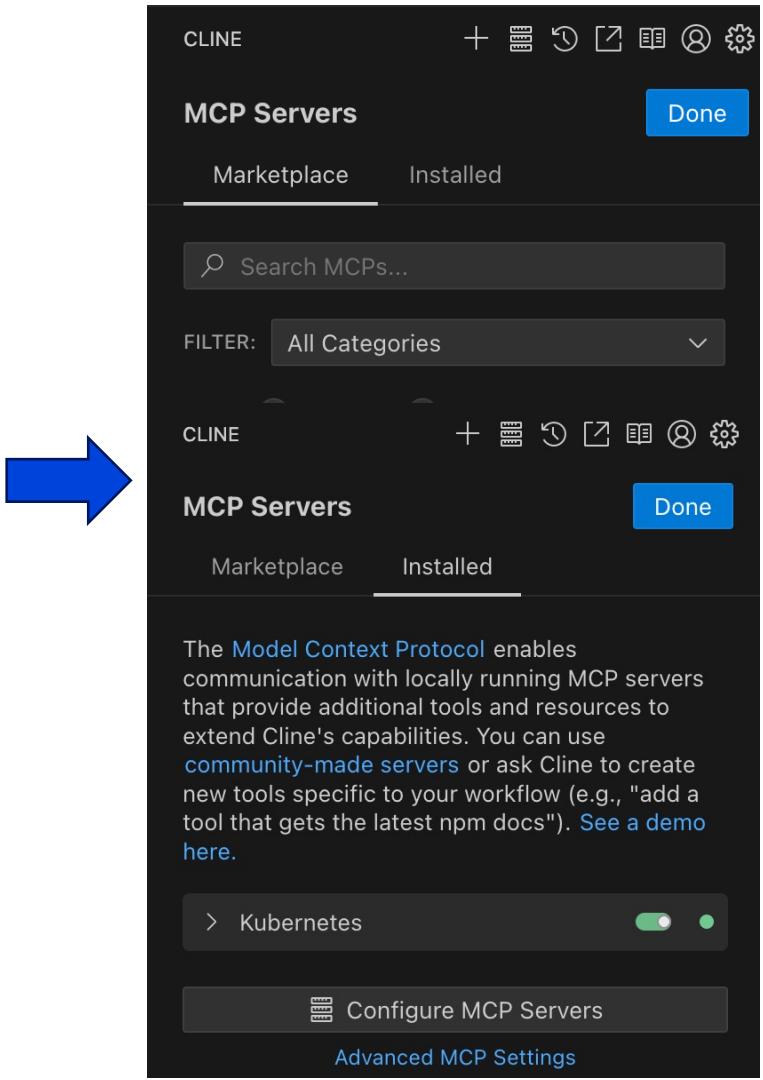
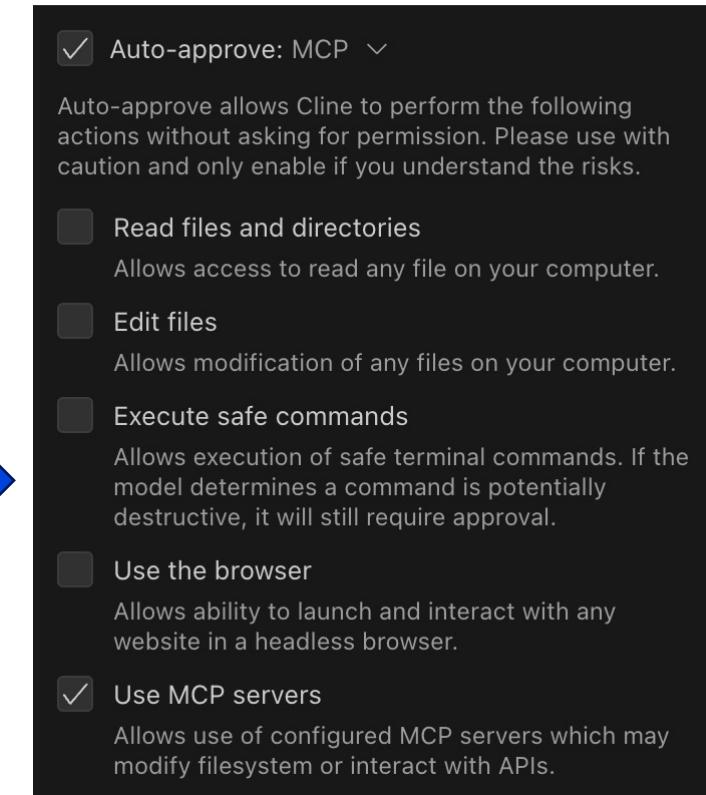
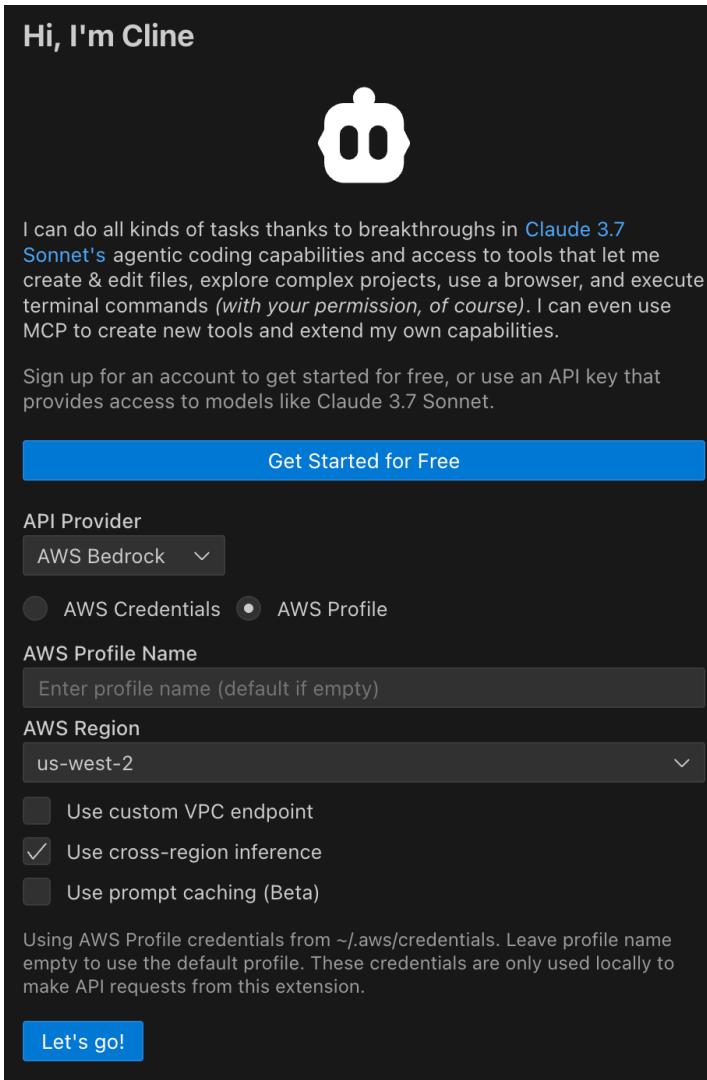
## MCP (Model Context Protocol)

- 애플리케이션이 LLM에게 컨텍스트를 제공하는 표준 프로토콜 (2024/11 Anthropic에 의해 도입)
- 각 데이터 소스 및 도구에 대한 사전 정의된 커넥터를 연결하거나 직접 커스텀으로 만든 MCP 서버를 연결할 수 있음
- <https://github.com/punkpeye/awesome-mcp-servers>



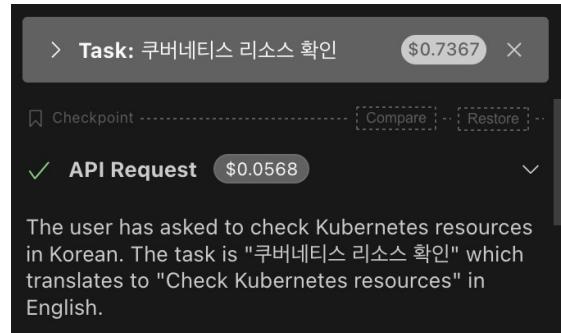
# MCP 활용 예시

## Cline과 Amazon Bedrock을 활용해 MCP 사용하기



# MCP 활용 예시

## Cline과 Amazon Bedrock을 활용해 MCP 사용하기



```
conversations, i.e. don't end your responses with questions or offers for further assistance.", "type": "text"}], "messages": [{"role": "user", "content": [{"type": "text", "text": "<task>\nкувернетес 리소스 확인\n</task>"}], {"type": "text", "text": "<environment_details>\n# VSCode Visible Files\n.../Library/Application Support/Code/User/globalStorage/saoudrizwan.claude-dev/settings/cline_mcp_settings.json\n# VSCode Open Tabs\\ntest.py\n.../Library/Application Support/Code/User/globalStorage/saoudrizwan.claude-dev/settings/cline_mcp_settings.json\n# Current Time\n3/24/2025, 4:45:17 PM (Asia/Seoul, UTC+9:00)\n# Current Working Directory (/Users/heuri/dev/mcp) Files\\ntest.py\n# Current Mode\nACT MODE\n</environment_details>"}]}, {"role": "assistant", "content": [{"type": "text", "text": "<thinking>\nThe user has asked to check Kubernetes resources in Korean. The task is \"кувернетес 리소스 확인\" which translates to \"Check Kubernetes resources\" in English.\nLooking at the environment details, I can see:\n1. The current working directory is `/Users/heuri/dev/mcp`\n2. There's a file called `test.py` in the directory\n3. The VSCode has open the MCP settings file at `.../Library/Application Support/Code/User/globalStorage/saoudrizwan.claude-dev/settings/cline_mcp_settings.json`\n4. We're in ACT MODE\nI notice that I have access to the Kubernetes MCP server (github.com/Flux159/mcp-server-kubernetes) as mentioned in the MCP SERVERS section of my system prompt. This server provides tools for interacting with Kubernetes clusters.\nTo check Kubernetes resources, I can use several tools provided by the MCP server:\n- list_namespaces: To list all namespaces\n- list_pods: To list pods in a namespace\n- list_deployments: To list deployments in a namespace\n- list_services: To list services in a namespace\n- list_nodes: To list all nodes in the cluster\nI'll start by checking if the Kubernetes cluster is accessible by listing the namespaces. This will give us a good starting point to explore other resources.\n</thinking>\nI'll check the Kubernetes resources in your cluster. Let me start by listing the available namespaces.\n<use_mcp_tool>\n<server_name>github.com/Flux159/mcp-server-kubernetes</server_name>\n<tool_name>list_namespaces</tool_name>\n<arguments>\n{}</arguments>\n</use_mcp_tool>"}]}]
```

pod중에서 pending 상태인거 왜그래?

**API Request** \$0.0618

The user's response in Korean translates to: "Why are some pods in pending state?"

This suggests that the user is seeing some pods in the Pending state and wants to know why. To investigate this, I need to:

1. Check which pods are in the Pending state
2. Describe those pods to get more details about why they're pending

Let's first get a list of all pods with their statuses to identify which ones are in the Pending state:

Let me check which pods are in the Pending state and investigate why:

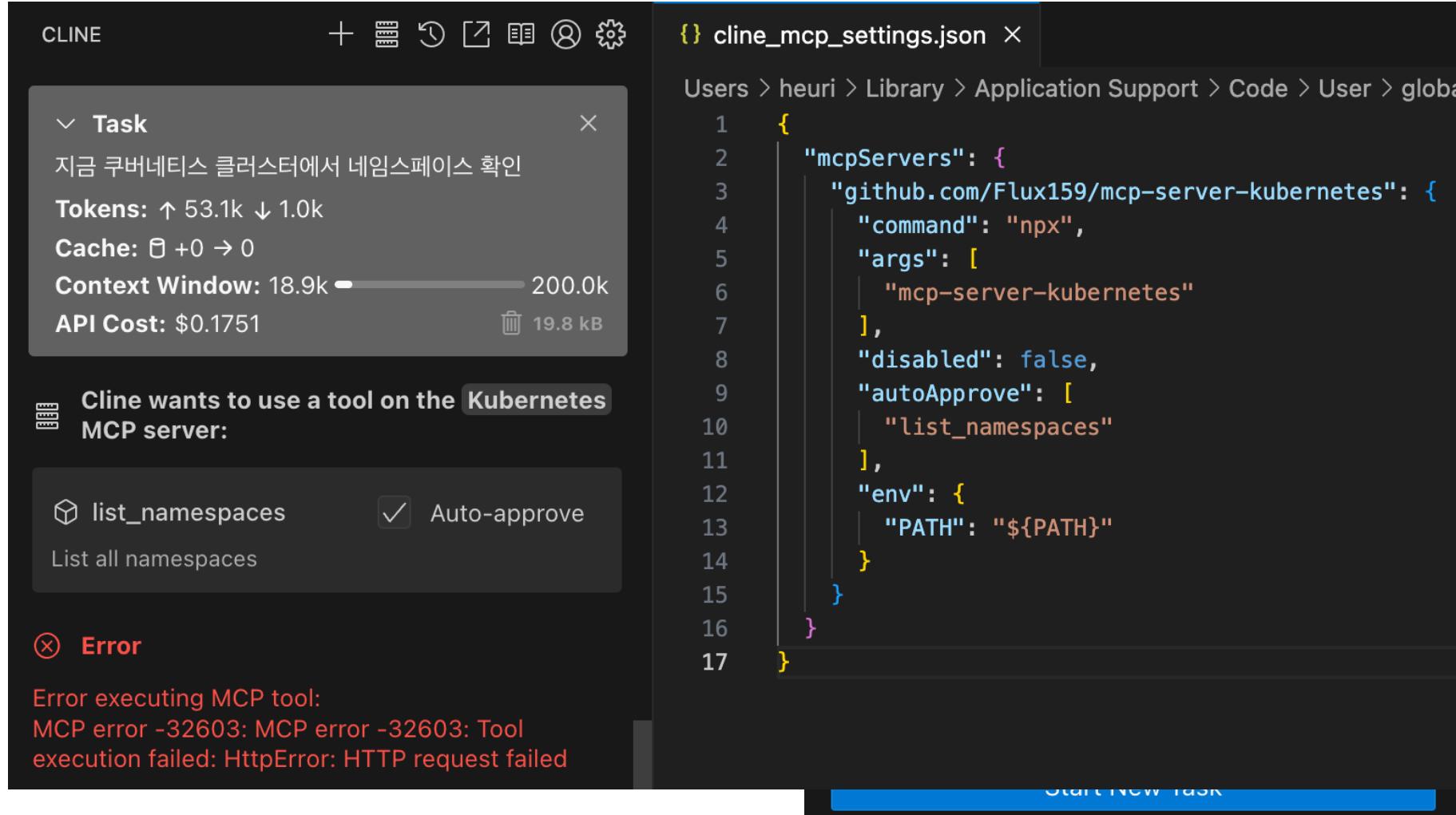
Cline wants to execute this command:

```
kubectl get pods --all-namespaces | grep Pending
```

Checkpoint ----- [Compare] [Restore]

# MCP 활용 예시

Cline과 Amazon Bedrock을 활용해 MCP 사용하기



The screenshot displays the Cline application interface and a code editor side-by-side.

**Cline Application:**

- Task Panel:** Shows a summary of resource usage:
  - Tokens: ↑ 53.1k ↓ 1.0k
  - Cache: 0 +0 → 0
  - Context Window: 18.9k — 200.0k
  - API Cost: \$0.1751
- Tool Selection:** A modal dialog titled "Cline wants to use a tool on the Kubernetes MCP server:" contains:
  - A checkbox for "list\_namespaces" which is checked.
  - An "Auto-approve" checkbox which is checked.
  - A button labeled "List all namespaces".
- Error Panel:** Displays an error message: "Error executing MCP tool: MCP error -32603: MCP error -32603: Tool execution failed: HttpError: HTTP request failed".

**Code Editor:**

```
{} cline_mcp_settings.json ×  
Users > heuri > Library > Application Support > Code > User > global  
1 {  
2   "mcpServers": {  
3     "github.com/Flux159/mcp-server-kubernetes": {  
4       "command": "npx",  
5       "args": [  
6         "mcp-server-kubernetes"  
7       ],  
8       "disabled": false,  
9       "autoApprove": [  
10         "list_namespaces"  
11       ],  
12       "env": {  
13         "PATH": "${PATH}"  
14       }  
15     }  
16   }  
17 }
```

**Bottom Bar:** A blue button labeled "Start New Task" is visible at the bottom right.

**K8s MCP**

K8s MCP

# Demo

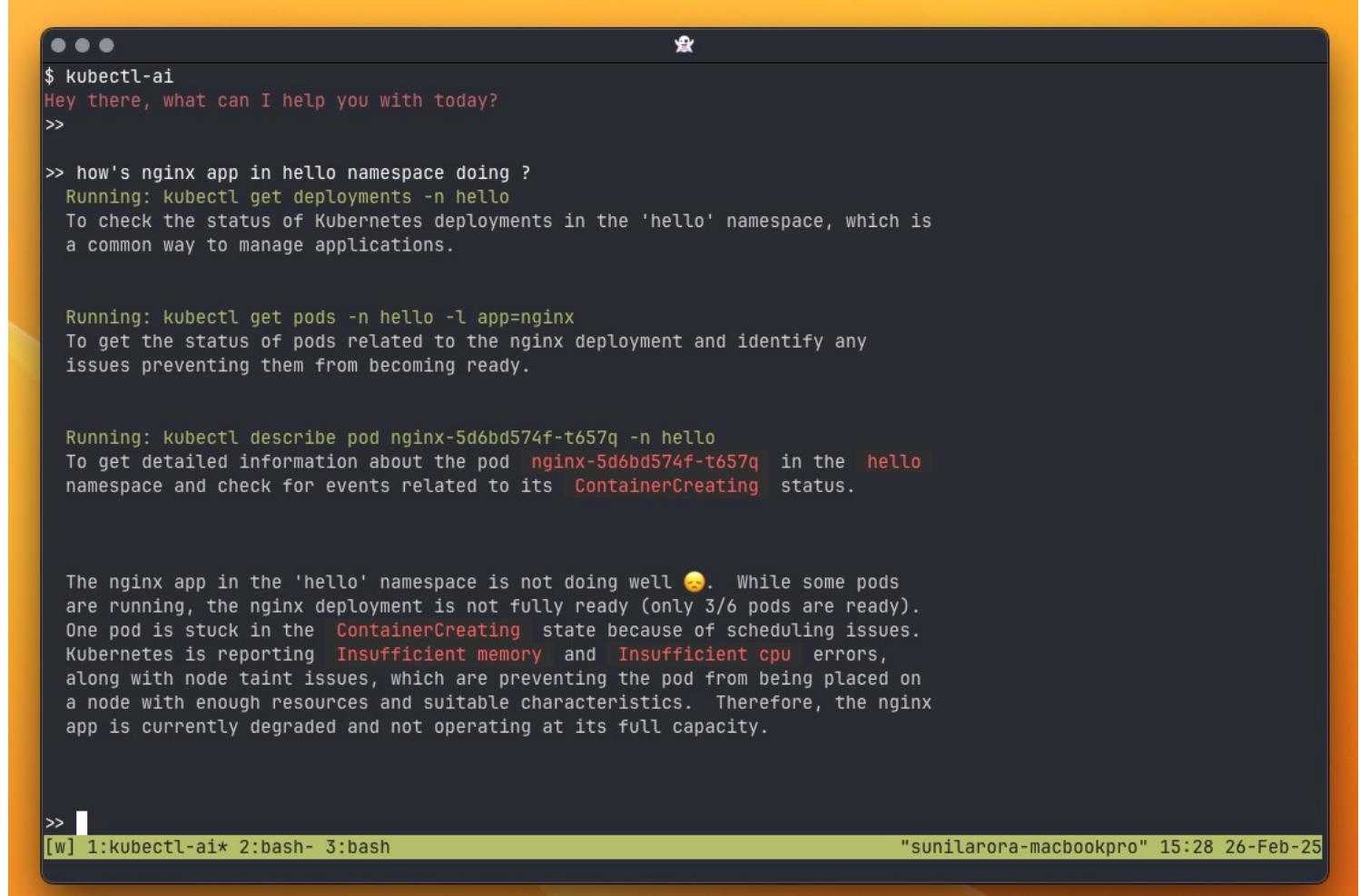
# kubectl-ai

## kubectl-ai

GoogleCloudPlatform – kubectl-ai

## kubectl-ai

- AI 기능을 활용하여 Kubernetes CLI 경험을 향상시키는 쿠버네티스 어시스턴트
- Gemini와 같은 대규모 언어 모델(LLM)을 활용하여 사용자가 Kubernetes 클러스터와 더 효율적으로 상호작용할 수 있도록 함



\$ kubectl-ai  
Hey there, what can I help you with today?  
>>  
  
=> how's nginx app in hello namespace doing ?  
Running: kubectl get deployments -n hello  
To check the status of Kubernetes deployments in the 'hello' namespace, which is a common way to manage applications.  
  
Running: kubectl get pods -n hello -l app=nginx  
To get the status of pods related to the nginx deployment and identify any issues preventing them from becoming ready.  
  
Running: kubectl describe pod nginx-5d6bd574f-t657q -n hello  
To get detailed information about the pod `nginx-5d6bd574f-t657q` in the `hello` namespace and check for events related to its `ContainerCreating` status.  
  
The nginx app in the 'hello' namespace is not doing well 😞. While some pods are running, the nginx deployment is not fully ready (only 3/6 pods are ready). One pod is stuck in the `ContainerCreating` state because of scheduling issues. Kubernetes is reporting `Insufficient memory` and `Insufficient cpu` errors, along with node taint issues, which are preventing the pod from being placed on a node with enough resources and suitable characteristics. Therefore, the nginx app is currently degraded and not operating at its full capacity.  
  
>>

[w] 1:kubectl-ai\* 2:bash- 3:bash "sunilarora-macbookpro" 15:28 26-Feb-25

# kubectl-ai

GoogleCloudPlatform – kubectl-ai

**kubectl-ai** on  main [!?] via  v1.24.1

```
→ kubectl-ai "AWSKRUG 네임스페이스 만들고, 최신 엔진엑스 1개 띄워줘"
→ k get po -n awskrug
NAME           READY   STATUS    RESTARTS   AGE
nginx-7584b6f84c-pq5tl   1/1     Running   0          2m28s
```

**kubectl-ai** on  main [!?] via  v1.24.1

```
→ kubectl-ai "awskrug에 새로 생성한 파드 잘 생성되었는지 확인"
```

사용자가 'awskrug' 네임스페이스에 새로 생성한 파드가 잘 생성되었는지 확인하는  
요청입니다. 'kubectl get pods -n awskrug' 명령어를 사용하여 해당 네임스페이스의  
파드 목록을 확인하고, 상태를 확인하여 파드가 정상적으로 생성되었는지 확인할 수  
있습니다.

```
Running: kubectl get pods -n awskrug
Are you sure you want to run this command (Y/n)?
```

Y

파드가 정상적으로 실행 중인 것으로 보입니다. 파드의 이름을 확인하고 상태가  
'Running'인지 확인했습니다. 'READY' 필드도 1/1로 표시되어 컨테이너가 모두  
준비되었음을 나타냅니다.

awskrug 네임스페이스에 생성하신 파드 (nginx-7584b6f84c-pq5t1)는 정상적으로  
생성되어  
실행 중입니다. 

# kubectl-ai

코드로 알아보는 – kubectl-ai (아직 문서가 없어요…)

## gollm/factory.go 45L ~ 56L

[kubectl-ai / gollm / factory.go](#)

Code

Blame

242 lines (205 loc) · 6.89 KB

```
32     func NewClient(ctx context.Context, providerID string) (Client, error) {
45         switch providerID {
46             case "gemini":
47                 return NewGeminiClient(ctx)
48             case "vertexai":
49                 return NewVertexAIClient(ctx)
50             case "ollama":
51                 return NewOllamaClient(ctx)
52             case "llamacpp":|
53                 return NewLlamaCppClient(ctx)
54             default:
55                 return nil, fmt.Errorf("unknown LLM_CLIENT scheme %q", providerID)
56         }
57     }
```

## 지원 모델

### 지원 프로바이더

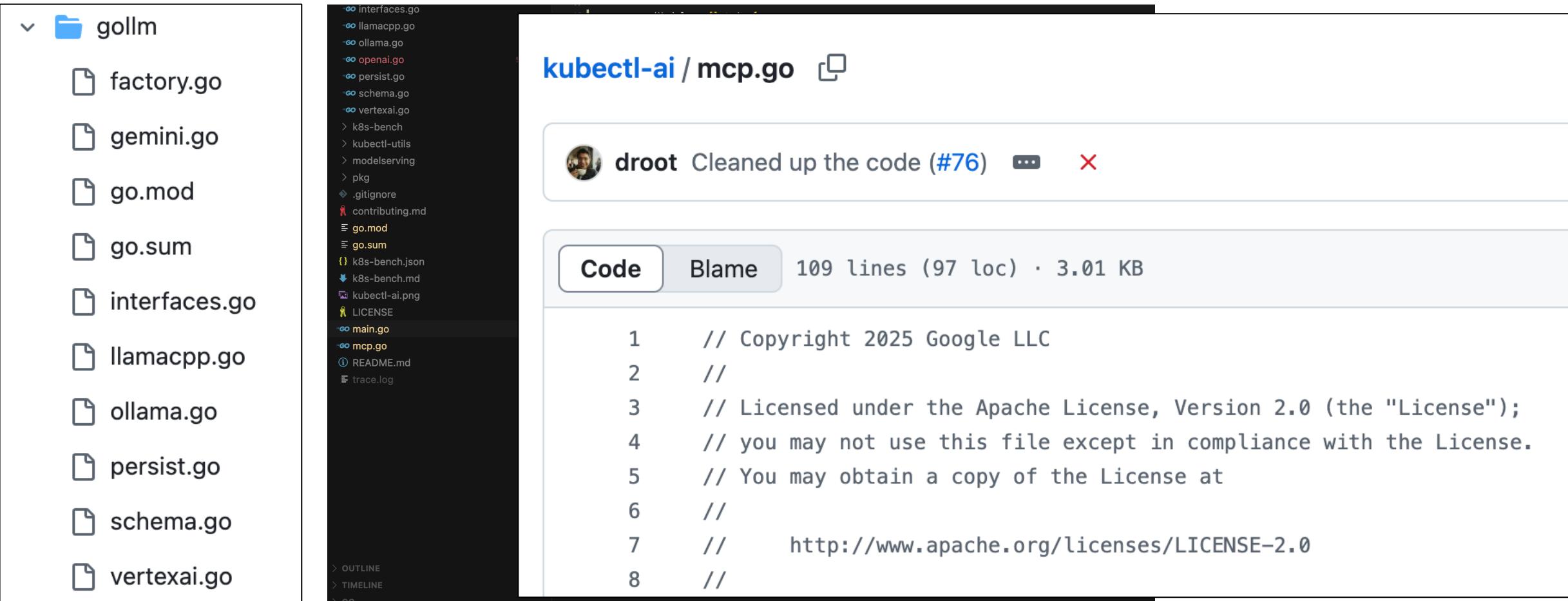
- gemini - Google의 Gemini 모델 (기본값)
- vertexai - Google Cloud의 Vertex AI
- ollama - 로컬에서 실행할 수 있는 오픈소스 모델 서비스
- llamacpp - llama.cpp를 기반으로 한 로컬 모델 실행

### 다른 모델 사용 방법

- kubectl-ai --llm-provider="ollama" \  
--model="llama3" "your query"

# kubectl-ai

Hmm...



The screenshot shows a file tree on the left and a code editor on the right.

**File Tree:**

- gollm
  - factory.go
  - gemini.go
  - go.mod
  - go.sum
  - interfaces.go
  - llamacpp.go
  - ollama.go
  - persist.go
  - schema.go
  - vertexai.go
- .gitignore
- contributing.md
- go.mod
- go.sum
- k8s-bench.json
- k8s-bench.md
- kubectl-ai.png
- LICENSE
- main.go
- mcp.go
- README.md
- trace.log

**Code Editor (kubectl-ai / mcp.go):**

**Commit:** droot Cleaned up the code (#76)

**Code Statistics:** 109 lines (97 loc) · 3.01 KB

```
1 // Copyright 2025 Google LLC
2 //
3 // Licensed under the Apache License, Version 2.0 (the "License");
4 // you may not use this file except in compliance with the License.
5 // You may obtain a copy of the License at
6 //
7 //     http://www.apache.org/licenses/LICENSE-2.0
8 //
```

**kubectl-ai Demo**

kubectl-ai Demo

# Demo

# 정리

# 대화형 쿠버네티스 비교

대화형 쿠버네티스 어디까지 왔나?

## 대화형 쿠버네티스 운영 범주

### K8sGPT

- podAnalyzer
- pvcAnalyzer
- rsAnalyzer
- serviceAnalyzer
- eventAnalyzer
- ingressAnalyzer
- statefulSetAnalyzer
- deploymentAnalyzer
- cronJobAnalyzer
- nodeAnalyzer
- mutatingWebhookAnalyzer
- validatingWebhookAnalyzer
- Optional
- hpaAnalyzer
- pdbAnalyzer
- networkPolicyAnalyzer
- gatewayClass
- gateway
- httproute
- logAnalyzer

### K8s MCP

-  create\_cronjob.ts
-  create\_deployment.ts
-  create\_namespace.ts
-  create\_pod.ts
-  delete\_deployment.ts
-  delete\_pod.ts
-  describe\_cronjob.ts
-  describe\_deployment.ts
-  describe\_pod.ts
-  get\_events.ts
-  get\_job\_logs.ts
-  get\_logs.ts
-  helm-operations.ts
-  kubectl-operations.ts
-  list\_cronjobs.ts
-  list\_deployments.ts
-  list\_jobs.ts
-  list\_nodes.ts
-  list\_pods.ts
-  list\_services.ts
-  port\_forward.ts
-  scale\_deployment.ts

### kubectl-ai (bench)

- ✓  k8s-bench
  - >  pkg
  - ✓  tasks
    - >  configure-ingress
    - >  create-network-policy
    - >  create-pod-mount-configmaps
    - >  create-pod-resources-limits
    - >  create-pod
    - >  fix-crashloop
    - >  fix-image-pull
    - >  fix-service-routing
    - >  list-images-for-pods
  - >  scale-deployment
  - >  scale-down-deployment

# 대화형 쿠버네티스 비교

대화형 쿠버네티스 어디까지 왔나?

## 프롬프트 비교 분석

분류	K8sGPT	kubectl-ai
분석 대상 프롬프트	pkg/ai/prompts.go	pkg/agent/systemprompt_template_default.txt
주요 목적	<ul style="list-style-type: none"> <li>특정 오류 메시지 해석 및 해결책 제시</li> </ul>	<ul style="list-style-type: none"> <li>종합적인 쿠버네티스 클러스터 관리 어시스턴트</li> </ul>
응답 형식	<ul style="list-style-type: none"> <li>간결한 텍스트(Error/Solution 구조)</li> </ul>	<ul style="list-style-type: none"> <li>JSON 구조</li> </ul>
응답 길이	<ul style="list-style-type: none"> <li>280 ~ 300자로 제한된 간결한 응답</li> </ul>	<ul style="list-style-type: none"> <li>제한 없음, 상세한 설명 가능</li> </ul>
자율성	<ul style="list-style-type: none"> <li>템플릿 기반의 정형화된 응답</li> </ul>	<ul style="list-style-type: none"> <li>다양한 해결책 탐색 및 자체 실행</li> </ul>
특화 기능	<ul style="list-style-type: none"> <li>특화된 오류 해석(Prometheus, Kyverno 등)</li> </ul>	<ul style="list-style-type: none"> <li>클러스터 리소스 직접 관리 및 조작</li> </ul>
접근 방식	<ul style="list-style-type: none"> <li>즉각적인 오류 해석 및 해결책 제시 (--interactive 옵션 미적용)</li> </ul>	<ul style="list-style-type: none"> <li>대화형 및 단계적 문제 해결</li> </ul>
사용자 상호작용	<ul style="list-style-type: none"> <li>최소한의 상호작용, 주로 일방향 응답 (--interactive 옵션 미적용)</li> </ul>	<ul style="list-style-type: none"> <li>대화 기반, 필요시 추가 정보 요청</li> </ul>
복잡성	<ul style="list-style-type: none"> <li>단순하고 직접적인 오류 해석</li> </ul>	<ul style="list-style-type: none"> <li>복잡한 추론 과정과 다단계 문제 해결</li> </ul>
이모지 사용	<ul style="list-style-type: none"> <li>언급 없음</li> </ul>	<ul style="list-style-type: none"> <li>일부 권장</li> </ul>
언어 지원	<ul style="list-style-type: none"> <li>다국어 지원 명시</li> </ul>	<ul style="list-style-type: none"> <li>언급 없음 (기본적으로 다국어 지원 기능)</li> </ul>

## 대화형 쿠버네티스 비교

kubectl-ai k8s-bench

### 추론을 통한 문제 해결?

#### Model Performance Summary

Model	chat-based Success	chat-based Fail	react Success	react Fail
gemini-2.0-flash	5	2	5	2
gemini-2.0-flash-thinking-exp-01-21	5	2	4	3
gemma-3-27b-it	5	1	4	2
<b>Total</b>	<b>15</b>	<b>5</b>	<b>13</b>	<b>7</b>

#### 1. chat-based 전략

- 전통적인 대화형(conversational) 방식을 사용
- LLM 모델에 직접적인 메시지를 보내고 응답을 받는 방식
- 함수 호출(function calling) 기능을 사용하여 도구 활용

#### 2. react 전략

- ReAct(Reasoning + Acting) 패턴 사용
- 모델이 구조화된 JSON 형식으로 응답하며, 사고(Thought), 응답(Answer), 행동(Action)을 포함
- 모델이 도구를 사용하기 전에 먼저 생각을 정리하고 이유를 설명하는 과정을 거침

# Q & A



감사합니다.