kagent と MCP で実現する Kubernetes-native な AIOps

近藤智文 / サイバーエージェント

CODT2026

自己紹介

近藤 智文

- サイバーエージェント24新卒
- プライベートクラウド IaaS 基盤の開発・運用をやってます
- OpenStack/OVS/Kubernetes /Go/Python

https://github.com/TOMOFUMI-KONDO/
https://x.com/tomokon_0314

本発表で学べること

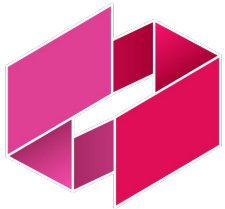
- kagent (AI Agent Runtime) の基本的な概念や使い方
- kagent を用いた AIOps の一例として、アラート自動応答エージェントシステムの設計パターンや具体的な実装

新リージョンにおける
アラート対応の課題

プライベートクラウドの概要

社内ユーザに以下の機能を提供

- IaaS (OpenStackベース)
- PaaS/SaaS
  - Kubernetes as a Service
  - Serverless Computing
  - ML Platform
  - GitHub Actions Self-Hosted Runner
  - DB/Storage, etc.



Cycloud

新リージョンの立ち上げ

ハードもソフトもゼロベースで構築

- 最新世代のハードウェアによりリソース効率やパフォーマンスの大幅向上
- 統合されたUI/UX・メンテナンスタイムの削減による利用者の負担削減
- 高いテナント分離性によるセキュリティ向上

課題：アラート対応による疲弊

新リージョンの立ち上げに伴いアラート対応が増大...

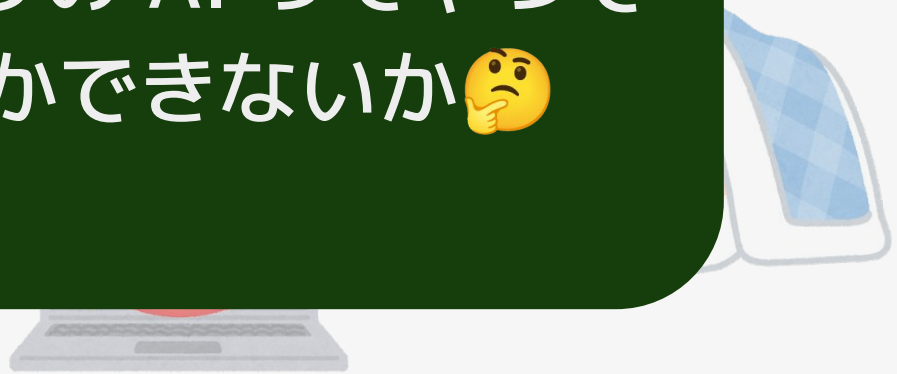
- インスタンスが起動しない
- HV (Hypervisor) のネットワークが全断
- エラーログの急増
- etc.

課題：アラート対応による疲弊

新リージョンの立ち上げに伴いアラート対応が増大...

- インス
- HV (Hy
- エラー
- etc.

人力対応が辛いので、
最近流行りの AI ってやつで
なんとかできないか🤔



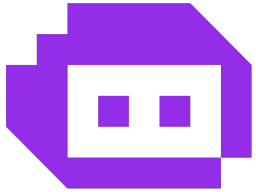
AI Agent による
アラート対応

LLM / AI Agent の発展

AIOps への広がり

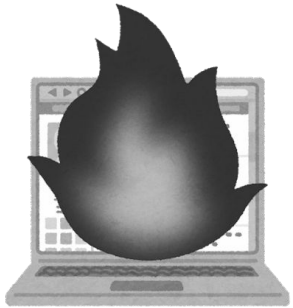
開発だけでなく、運用にも AI Agent を導入する
ソリューションやユースケースが増えている。

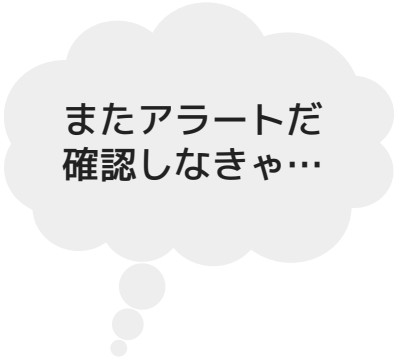
- Datadog Bits AI
- AWS DevOps Agent
- kagent
- etc.



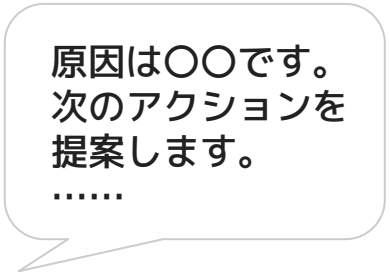
やりたいこと

従来のアラート対応





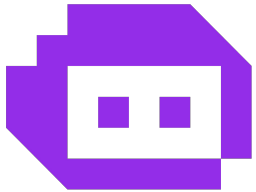
AIエージェントを組み込んだ アラート対応



AIOps への広がり

今回は、AIOps のソリューションとして kagent に着目

- Datadog Bits AI
- AWS DevOps Agent
- kagent←←←
- etc.



kagent

- CNCF Sandbox Project の OSS
- **k8s-native** な AI Agent Runtime
  - 全てのリソースを **k8s リソースとして定義**
  - k8s やエコシステムの調査・操作のツールが標準搭載
  - 組み込みの GUI や CLI も搭載
- **A2A / MCP 連携機能**
- 豊富な LLM Provider 対応

アラート自動応答
エージェントシステム

システムフロー

アラート発生

検知

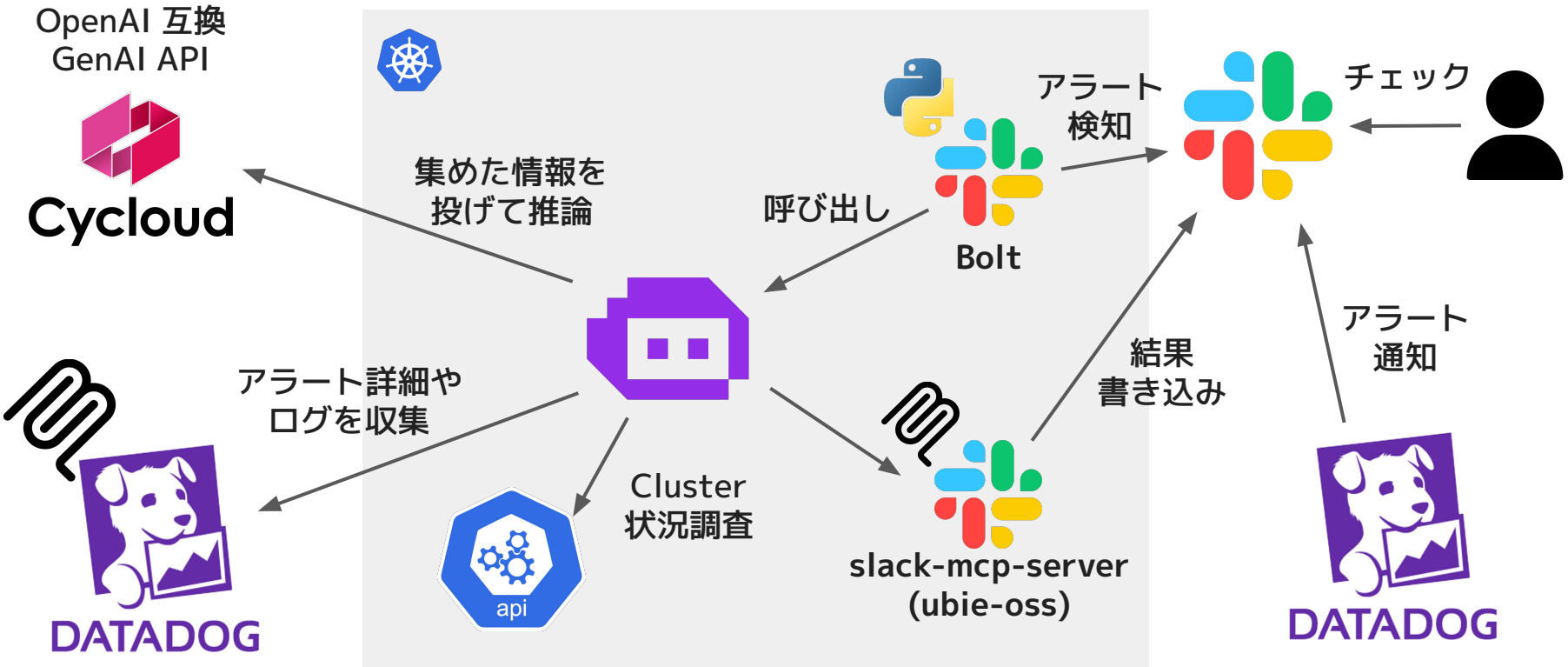
調査

判断

報告

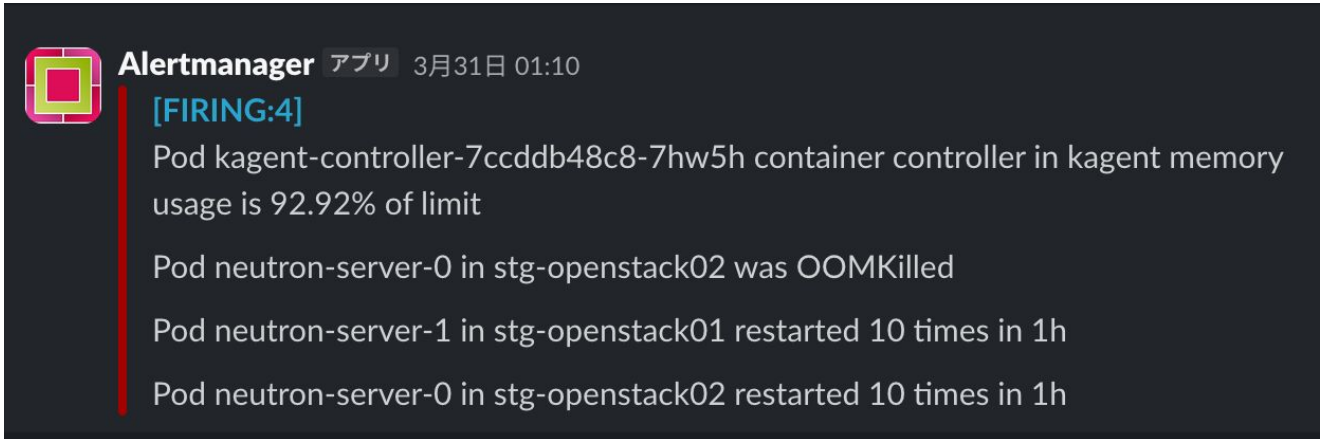
確認・最終判断

システム設計


OpenAI 互換
GenAI API
Cycloud
集めた情報を
投げて推論
呼び出し
Bolt
アラート
検知
チェック
アラート詳細や
ログを収集
Cluster
状況調査
api
結果
書き込み
slack-mcp-server
(ubie-oss)
アラート
通知
DATADOG
DATADOG

アラートの検知

アラート発報🔥🔥🔥


Alertmanager アプリ 3月31日 01:10
[FIRING:4]
Pod kagent-controller-7ccddb48c8-7hw5h container controller in kagent memory usage is 92.92% of limit
Pod neutron-server-0 in stg-openstack02 was OOMKilled
Pod neutron-server-1 in stg-openstack01 restarted 10 times in 1h
Pod neutron-server-0 in stg-openstack02 restarted 10 times in 1h

Slack Bolt ソケットモードでアラート検知

Bolt

1. websocket 接続

2. 待機

3. アラート通知

4. アラートメッセージを送信

Slack Bolt ソケットモードでアラート検知

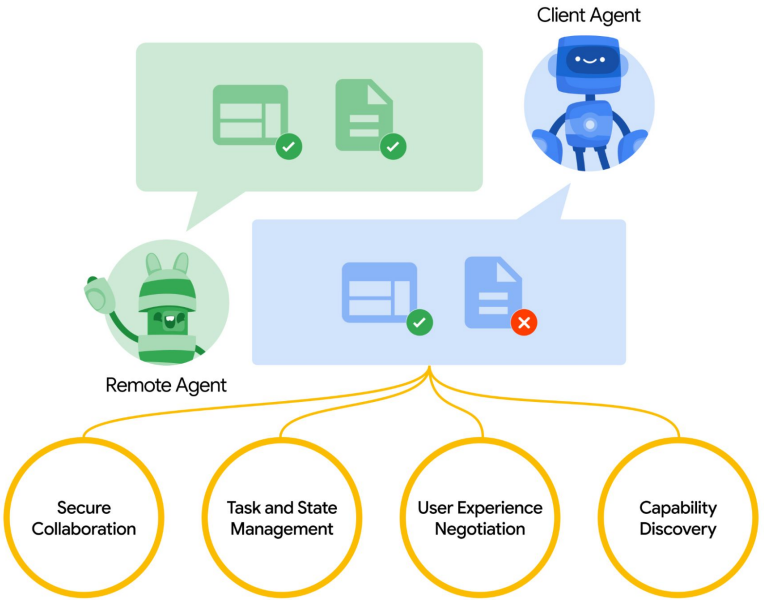
```
@app.event("message")
def handle_message(event, logger):
    channel = event.get("channel")
    if channel not in CHANNEL_IDS:
        return
    # Ignore thread replies to prevent duplicate triggers
    if event.get("thread_ts"):
        return

    msg_ts = event["ts"]
    msg_body = _extract_message_body(event)
    prompt = f"チャンネル {channel} のタイムスタンプ {msg_ts} のアラートを調査して、結果をスレッドに投稿してください。"
    if msg_body:
        prompt += f"\n\nアラート本文:\n{msg_body}"

    payload = ...
```

A2A による
Agent 呼び出し

A2A (Agent2Agent) Protocol

Google が提唱した
オープンスタンダードな
AI Agent 間通信プロトコル

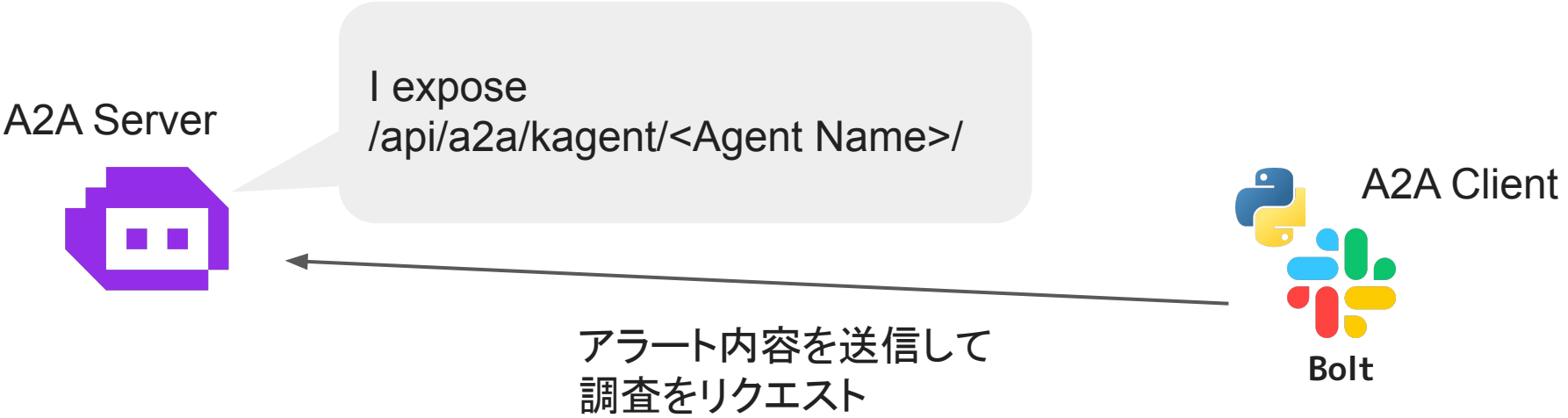
- Agent Discovery
- Invoke another agent
- Auth



https://developers.googleblog.com/en/a2a-a-new-era-of-agent-interoperability/

kagent & A2A

“Agent” リソースを作成するだけで A2A が利用可能



kagent & A2A

```
payload = json.dumps(
    {
        "jsonrpc": "2.0",
        "id": str(uuid.uuid4()),
        "method": "message/send",
        "params": {
            "message": {
                "role": "user",
                "messageId": str(uuid.uuid4()),
                "parts": [{"kind": "text", "text": prompt}],
            }
        },
    }
).encode()

req = urllib.request.Request(
    A2A_ENDPOINT,
    data=payload,
    headers={"Content-Type": "application/json"},
    method="POST",
)
```

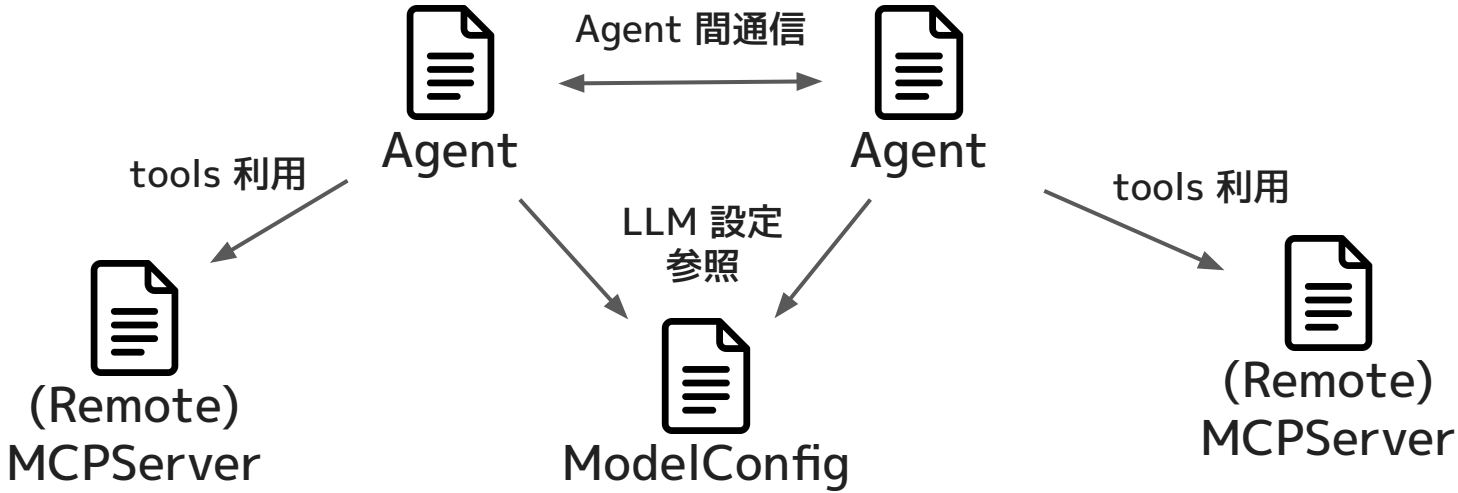
調査開始を報告


Alertmanager アプリ 3月31日 01:10
[FIRING:4]
Pod kagent-controller-7ccddb48c8-7hw5h container controller in kagent memory usage is 92.92% of limit
Pod neutron-server-0 in stg-openstack02 was OOMKilled
Pod neutron-server-1 in stg-openstack01 restarted 10 times in 1h
Pod neutron-server-0 in stg-openstack02 restarted 10 times in 1h
1 件の返信
Compute Alert Responder アプリ 3月31日 01:10
⏳ アラートを確認中です...

kagent における Agent 設定

Manifest (yaml) で全てを定義

kagent では Agent を含むあらゆるものを Kubernetes の Resource として定義することが可能

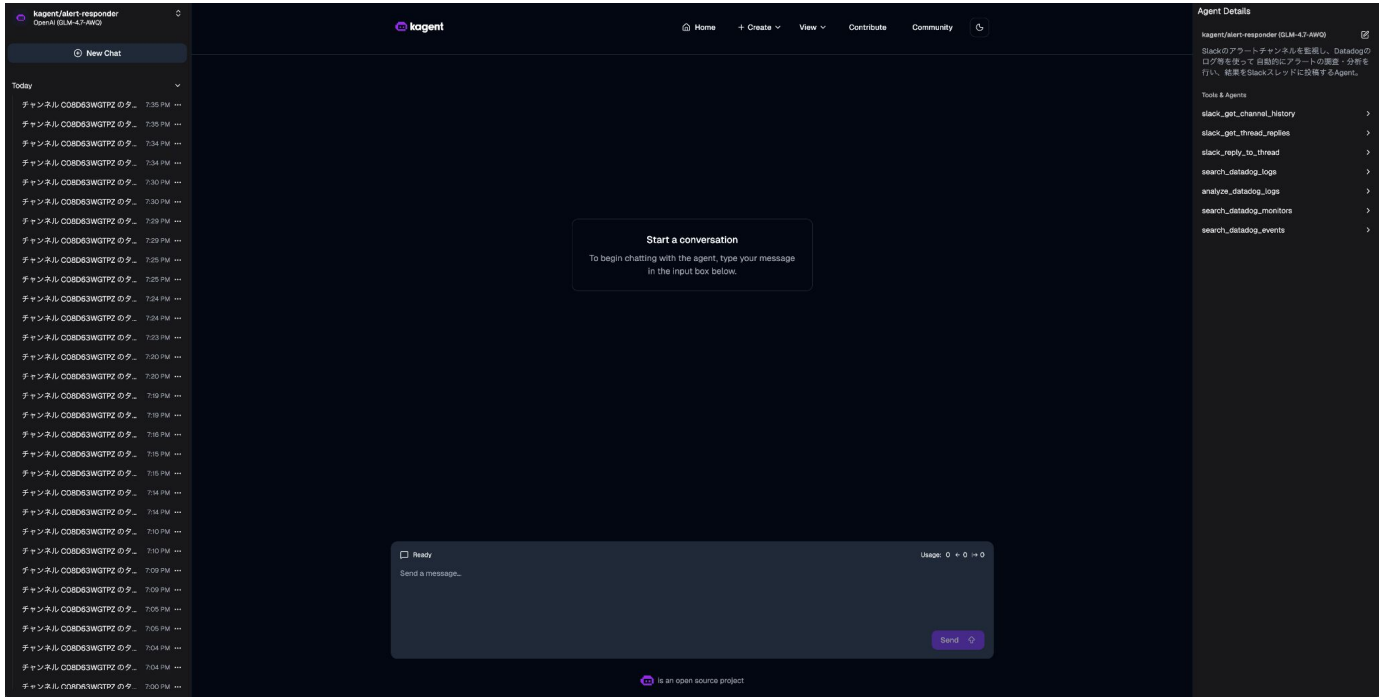


Agent Spec 詳細

```
spec:
  description: >
    Slackのアラートチャンネルを監視し、Datadogのログ等を使って
    自動的にアラートの調査・分析を行い、結果をSlackスレッドに投稿するAgent。
  declarative:
    systemMessage: |
      cycloudインフラチームのアラート対応エージェント。
      Slackのアラートを調査し、Datadogのモニタリングデータで分析を行う。
      すべての出力は日本語で、敬語を使わず簡潔に書くこと。

      ## ワークフロー

      ### Step 0: 対応開始の通知
      - 調査を始める前に、元のアラートメッセージのスレッドに対応開始メッセージを投稿する
      - メッセージ例: ":hourglass_flowing_sand: アラートを確認中です…"

      ### Step 1: アラートの確認
      …

      ## ガイドライン
      …
```

Tips: kagent GUI

自動応答システムにおいて必須ではないが、組み込みの GUI からも Chat 形式で Agent を利用可能



ModelConfig

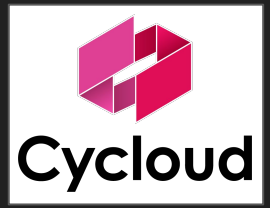
多くの LLM providers をサポートしており、
ModelConfig リソースを作成するだけで利用可能！

- Anthropic
- OpenAI
- Amazon Bedrock
- Google Vertex AI
- Gemini
- Azure OpenAI
- Local LLM (e.g. Ollama)
- etc.

ModelConfig

OpenAI 互換の 社内 AI サービス も簡単に利用可能！

```
apiVersion: kagent.dev/v1alpha2
kind: ModelConfig
metadata:
  name: cycloud-genai
spec:
  provider: OpenAI
  model: GLM-4.7-AWQ
  apiKeySecret: kagent-credentials
  apiKeySecretKey: GENAI_API_KEY
  openAI:
    baseUrl: "https://<Cycloud GenAI API の独自ドメイン>/v1"
```



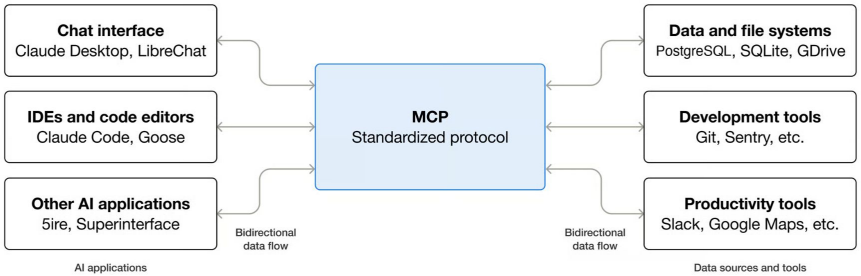
kagent と MCP

MCP (Model Context Protocol)

AI と外部ツール間のOpen Standard な通信プロトコル

e.g.

- モニタリングデータの取得
- Slack のメッセージ閲覧や送信
- GitHub 上のソースコードやコミットログを取得
- etc.



https://modelcontextprotocol.io/docs/getting-started/intro

kagent & MCP

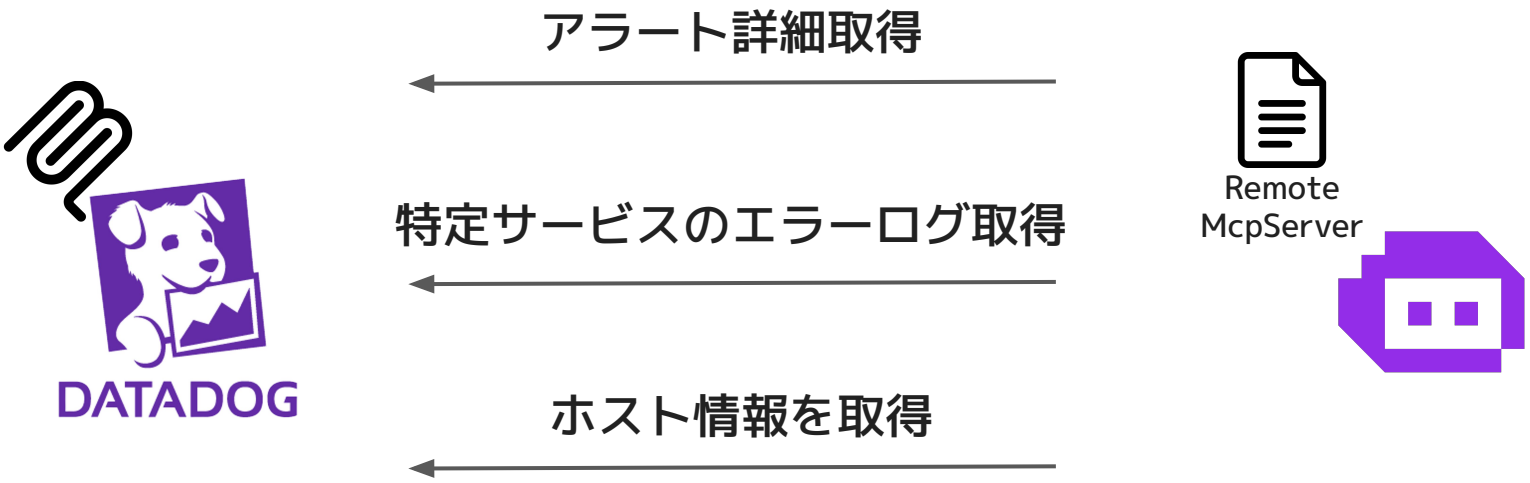
“RemoteMCPServer” リソースの作成と Agent 側での tools の指定だけで MCP server が利用可能！

```
apiVersion: kagent.dev/v1alpha2
kind: RemoteMCPServer
metadata:
  name: datadog-mcp
spec:
  description: "Datadog official remote MCP
server"
  protocol: STREAMABLE_HTTP
  url: "https://mcp.datadoghq.com/..."
  headersFrom:
    - name: ⋯
  timeout: 30s
```

```
spec:
  declarative:
   tools:
     - type: McpServer
       mcpServer:
         kind: RemoteMCPServer
         name: datadog-mcp
         toolNames:
           - search_datadog_logs
           - analyze_datadog_logs
           - search_datadog_monitors
           - search_datadog_events
```

kagent & Datadog MCP

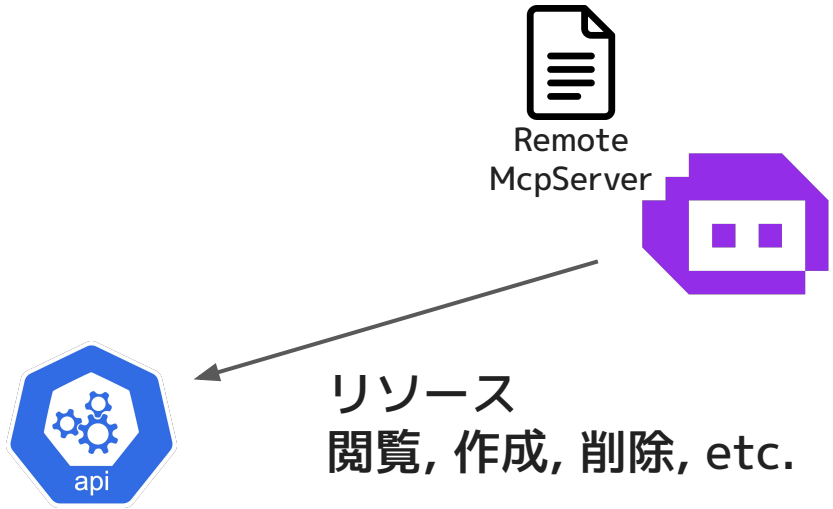
柔軟なクエリで Datadog 上の様々なモニタリングデータを閲覧・分析可能！



Datadog MCP 呼び出しの様子

kagent & Kubernetes tools MCP

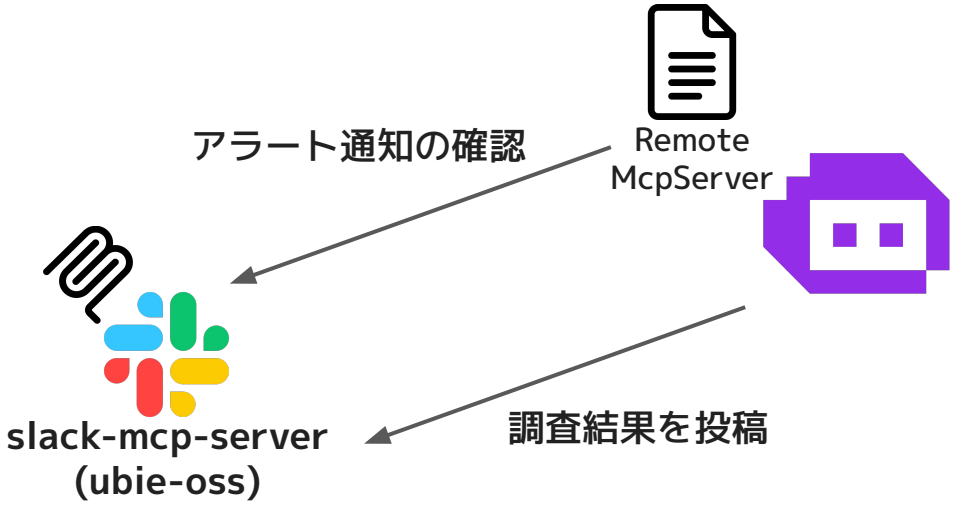
組み込みの Kubernetes tools が用意されており、標準セットアップで Agent から Cluster を操作可能




```
spec:
  declarative:
    tools:
    - mcpServer:
        apiGroup: kagent.dev
        kind: RemoteMCPServer
        name: kagent-tool-server
        toolNames:
        - k8s_check_service_connectivity
        - k8s_patch_resource
        - k8s_remove_annotation
        - k8s_annotate_resource
        …
      type: McpServer
```

kagent & slack-mcp-server (ubie-oss)

Agent から Slack のメッセージを閲覧・投稿




```
spec:
  declarative:
      tools:
      - type: McpServer
        mcpServer:
          kind: RemoteMCPServer
          name: slack-mcp
          toolNames:
            - slack_get_channel_history
            - slack_get_thread_replies
            - slack_reply_to_thread
```

slack-mcp-server を用いた調査結果の報告

Alertmanager アプリ 3月31日 00:55

[FIRING:4]

Pod kagent-controller-7ccddb48c8-7hw5h container controller in kagent memory usage is 92.07% of limit

Pod neutron-server-0 in stg-openstack02 was OOMKilled

Pod neutron-server-1 in stg-openstack01 restarted 10 times in 1h

Pod neutron-server-0 in stg-openstack02 restarted 11 times in 1h

2 件の返信

Compute Alert Responder アプリ 3月31日 00:55

⏳ アラートを確認中します...

🤖 **アラート分析**

サービス: kagent-controller, neutron-server
環境: stg
調査期間: 2026-03-30T14:54:23Z - 2026-03-30T15:54:23Z

概要:
neutron-serverが継続的にメモリ不足で再起動しており、クラスタ全体の空きメモリが枯渇している。kagent-controllerはメモリリミットの92.07%使用だが安定稼働中。

原因分析: (確度: 高)
neutron-serverのコンテナがメモリ制限を超過してOOMKilledされ、KubernetesスケジューラはInsufficient memoryエラーでPodの再配置に失敗している。clusterのノードごとに複数のworkerがメモリ不足になっているため、クラスター全体の空きメモリが枯渇。

根拠:
- `Task ran out of memory` イベントが過去1時間に20件以上発生（neutron-server-0, neutron-server-1）
- ノード stg-k8snode003 で `FailedScheduling` のエラーが継続: `0/4 nodes are available: 1 node(s) had untolerated taint {node-role.kubernetes.io/control-plane: }, 3 Insufficient memory`
- 最も新しいOOMイベント: `2026-03-30T15:54:23Z` - neutron-server-0 in stg-openstack02

推奨アクション:
1. クラスタのノードのメモリ使用状況を確認する（kubectl top node）
2. neutron-serverのmemory limitを引き上げる、またはメモリリークの原因調査
3. kagent-controllerのメモリリミットを引き上げる（現状92%使用）

課題と今後の展望

Agent による自動的な解決や HITL のフロー構築

- 原因の推定とネクストアクションの提示にしか至っておらず、実際の解決のためのアクションは人間が行う必要がある
  - 適切なシステムプロンプトと tools を渡せば Agent によるアクション実行は可能だが、内容によっては人間の承認を挟みたい
  - 適切な HITL (Human In The Loop) のフローを導入することで **Agent の信頼性を確保**することが求められる

データソースの拡充

- Kubernetes Cluster 情報 と Datadog の log は取れているが、Agent の精度を上げるために他にも様々なデータソースを取り入れ、活用したい
  - Metrics (grafana-mcp)
  - GitHub 上のソースコードやコミットログ
  - etc.

他チームとの連携

- 各チームでそれぞれ AI Agent Platform を構築している状態なので、それらを連携することで更なる効果が期待できる
  - VM の上で動いているマネージドサービスの障害時に、マネージドサービス側の Agent から IaaS 基盤側の Agent を経由して状態を取得するなど

まとめ

本発表での学び

- kagent により Agent や MCP 等あらゆるリソースを Kubernetes リソースとして定義し、宣言的な設定でお手軽に AI Agent Platform を構築する手法
- kagent による AIOps の一例として、アラート自動応答システムの設計パターン

kagent ぜひ使ってみてください！

We are hiring!

一緒に働く仲間を大募集しています！

- プライベートクラウドの開発・運用に興味がある方
- AI Agent Platform の構築や AIOps による業務効率化に取り組んでみたい方

お気軽にお声がけください 👋

https://it.cyberagent.group/team/ciu/
https://www.cyberagent.co.jp/way/list/detail/id=31427
https://www.cyberagent.co.jp/way/list/detail/id=31367
https://www.cyberagent.co.jp/way/list/detail/id=29497