ZABBIX 2021 Conference JAPAN

NTT Communications

コンテナでの監視機能の実装とproblemテーブル肥大化問題の事例紹介

2021年11月18日

エヌ・ティ・ティ・コミュニケーションズ株式会社

名倉　堂心（なくら　たかみ）

自己紹介

名倉 堂心（なくらたかみ）

【経歴】

2020年4月　　新卒でNTTコムソリューションズ入社（入社2年目）

2020年9月～　Zabbixの構築・運用業務に従事

2021年7月　　NTTコミュニケーションズ合併後もZabbixに関わる

【スポーツ歴】

サッカー歴9年、空手歴9年、ラグビー歴8年目（継続中）

会社紹介

NTT コミュニケーションズ株式会社

(https://www.ntt.com/)

・2008年より、Zabbix社と提携したZabbix関連事業を開始

・ZABICOMソリューションの導入/運用/製品供給などのサービスを提供

https://www.zabicom.com

目次

1. コンテナでの監視機能の実装について
2. problemテーブル肥大化に関する事例と対応について
3. ちょっとだけ宣伝
4. まとめ

コンテナでの監視機能の実装について

どんな要件があるのか？

要件

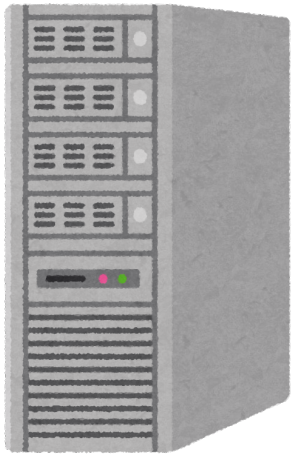
①ネットワーク・システムの監視をしたい

②オンプレ、仮想環境の監視にZabbixを導入したい

監視システムとして、Zabbixを

物理サーバまたは仮想サーバで導入するケースがほとんど

Zabbixの導入時に用いるプラットフォームは？



物理マシン

仮想マシン

コンテナ

今回お話するケースでは…

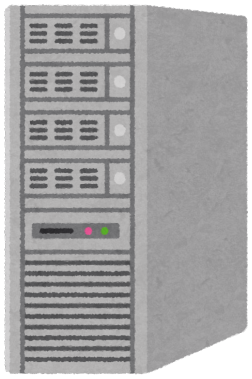
検証環境でOKが出たものを、本番環境に適用したい

という要件も…

検証環境から本番環境へ移植・適用する作業が必要

移植・適用のしやすさを考えると…

物理マシン



半日～1日

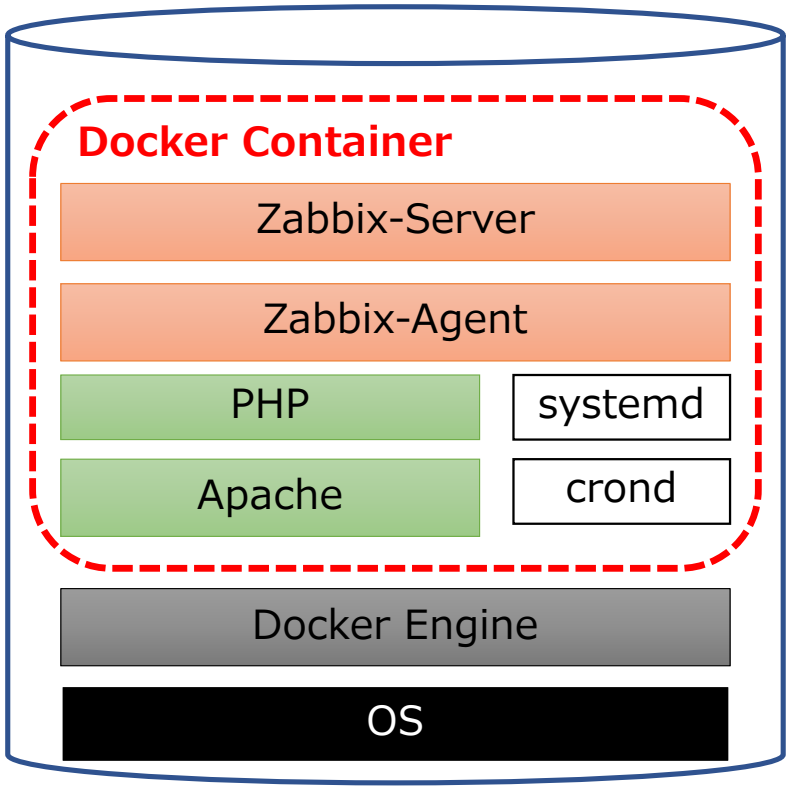
仮想マシン

2～3時間

コンテナ

5～15分

コンテナと言えど採用したのは…



サービスを1つのコンテナにまとめた構成

（All in Oneコンテナ）

ギモン＆ツッコミ

コンテナ使うのに
なんでひとまとめにしてんの？

オンプレでの設計と
変わんなくない？

コンテナ使う意味ww

A.ごもっともです。。。

仮にマイクロサービス化した場合

で、保守・運用どうするの？

たしかに(;'∀')
実績もないしどうしよう。。。

All in One構成にするメリット

1台に複数のマイクロサービスコンテナでZabbixを構築する場合との比較

メリット

✓オンプレとほぼ同様の構築手順

✓オンプレ構築と変わらない保守性、運用性

自分たちが保守・運用をするにあたっても確実な手段

つまるところ…

物理マシン

仮想マシン

運用実績の多さ

いいとこどり◎

コンテナ

持ち運びのしやすさ

まとめると

✓ 検証環境から本番環境への適用がスムーズ

✓ 自分たちが保守・運用するにあたっても安心

All in Oneコンテナが最適だった

problemテーブル肥大化に関する事例と対応について

problemテーブルとは？

データベース内の「障害中」のイベントデータを格納する場所（テーブル）

「監視データ」→「障害」画面の**最近の障害**または**障害**が参照するテーブル

Zabbix4.0以降の「正常イベント」の考え方

「絶賛障害中のイベントが復旧したよ！」が正常イベントの定義

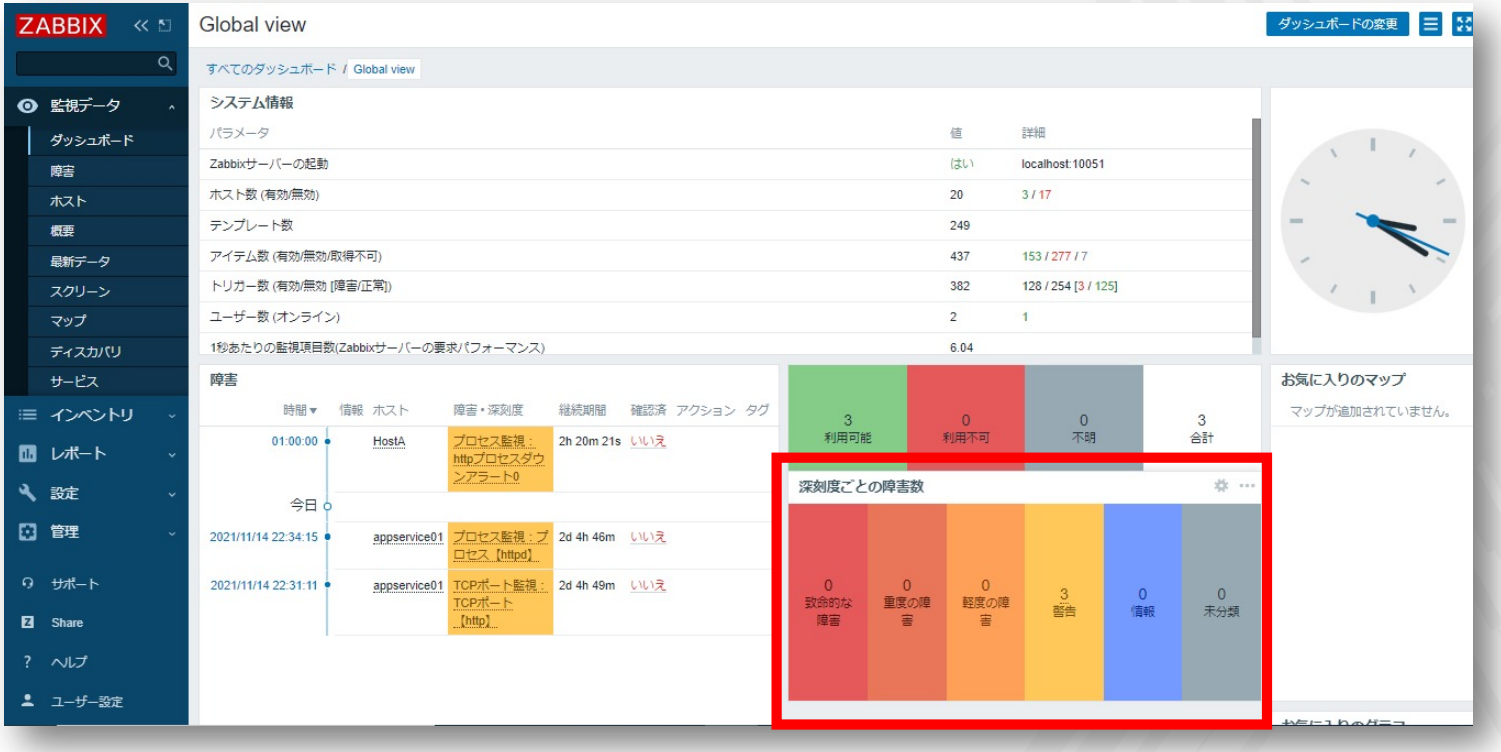
障害中のデータは復旧しない限り

データの保存期間を過ぎてもテーブルに残り続ける

「problemテーブル肥大化問題」ってナニ？

障害イベントが蓄積することで

ZabbixのWeb画面の表示が遅くなる、あるいは、表示できなくなる問題



problemテーブルが肥大化する原因

復旧しない障害イベントが蓄積すること

- ✓ イベント生成モード：複数
- ✓ 手動クローズを行わない運用

原因になりやすい監視

- ✓ **ログ監視**
- ✓ **SNMPTrap監視**

problemテーブルを肥大化させないために

- ✓ **復旧イベントを生成する**
- ✓ **定期的に手動クローズを行う**

復旧条件式の設定方法（regexp関数を使用する際）

監視するログファイル

監視するログファイルに存在する正規表現

障害条件式 {HostA:log[/var/log/messages,ERROR].regexp(ERROR,**30**)}= 1

アイテムキー

regexp関数：最新の値に第1引数の正規表現が、直近の第2引数に指定した期間内に存在するか判定する関数

復旧条件式 {HostA:log[/var/log/messages,ERROR].nodata(30)} = 1

nodata関数：特定の期間内にアイテムを取得してないかどうかを判定する関数

アイテム：監視データ取集の定義

トリガー：アイテムの障害判定の定義

regexp関数の第2引数がなぜ必要になるのか

障害条件式 {HostA:log[/var/log/messages,ERROR].regexp(ERROR,**30**)}= 1

復旧条件式 {HostA:log[/var/log/messages,ERROR].nodata(30)} = 1

実際に出会った事象

problemテーブル内にデータが200万件以上溜まり

Web画面にアクセスできなくなる

DBから直接データの削除を試みるも事態は好転せず…

DBの作り直し

```
MySQL [zbx0003je]> select count(*) from problem where r_clock=0 ;
+----------+
| count(*) |
+----------+
|  2441611 |
+----------+
1 row in set (5.844 sec)
```

どう対応したのか

一定期間データを取得しない場合に、自動で復旧させる条件式を設定

nodata関数を使用

現在は正常に運用ができている◎

自動復旧条件式の設定のしかた注意！

ログ監視で、第2引数を指定しないで
アイテムを取得することはよくありますよね？

アイテム: log[/var/log/messages]

トリガー:{HostA:log[/var/log/messages].regexp(**“ERROR”**)} = 1

自動復旧条件式の設定のしかた注意！

同じアイテムに複数のトリガーを設定する場合

トリガー①

障害条件式 {HostA:log[/var/log/messages].regexp(“**ERROR**”,30)}=1

復旧条件式 {HostA:log[/var/log/messages].nodata(30)} = 1

トリガー②

障害条件式 {HostA:log[/var/log/messages].regexp(“**CRITICAL**”,30)}=1

復旧条件式 {HostA:log[/var/log/messages].nodata(30)} = 1

どんなことが起こるか？

2021/11/18 **12:00:00** → ERROR → ERROR の障害を検知

2021/11/18 **12:00:01** → ERROR → ERROR の障害を検知

2021/11/18 **12:00:02** → CRITICAL → CRITICALの障害を検知

ERRORの障害も検知

原因は何？？

トリガー①

障害条件式 {HostA:log[/var/log/messages].regexp(“**ERROR**”,**30**)} = 1

トリガー②

障害条件式 {HostA:log[/var/log/messages].regexp(“**CRITICAL**”,**30**)}=1

2021/11/05 12:00:00 “ERROR”

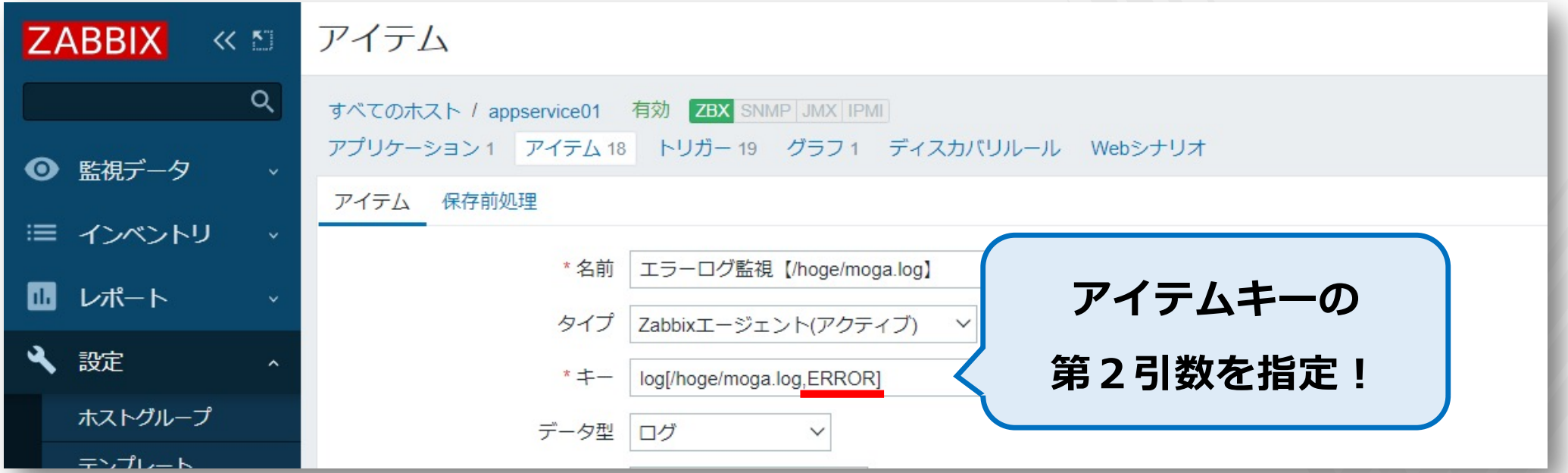
2021/11/05 12:00:01 “ERROR”

2021/11/05 12:00:02 “CRITICAL”

直近の30秒間に

“ERROR”アルネ

このような誤検知が起きないようにするために



データ収集の段階で、取り込むデータを絞り込む！

アイテムキーの第2引数を設定すると…

障害条件式

① {HostA:log[/var/log/messages,”**ERROR**”].regexp(“ERROR”,30)}=1

② {HostA:log[/var/log/messages,”**CRITICAL**”]. regexp(“CRITICAL”,30)}=1

①と②はそれぞれ異なるアイテムに紐づいている

誤検知を防ぐことができる！

忘れてはいけないのは…

目的はproblemテーブルを肥大化させないこと

あくまでも復旧条件式の設定は1つの手段

ちょっと宣伝

手動での障害復旧って大変ですよね…

✓特にログやTrapのバースト時は確認や復旧に時間がかかる

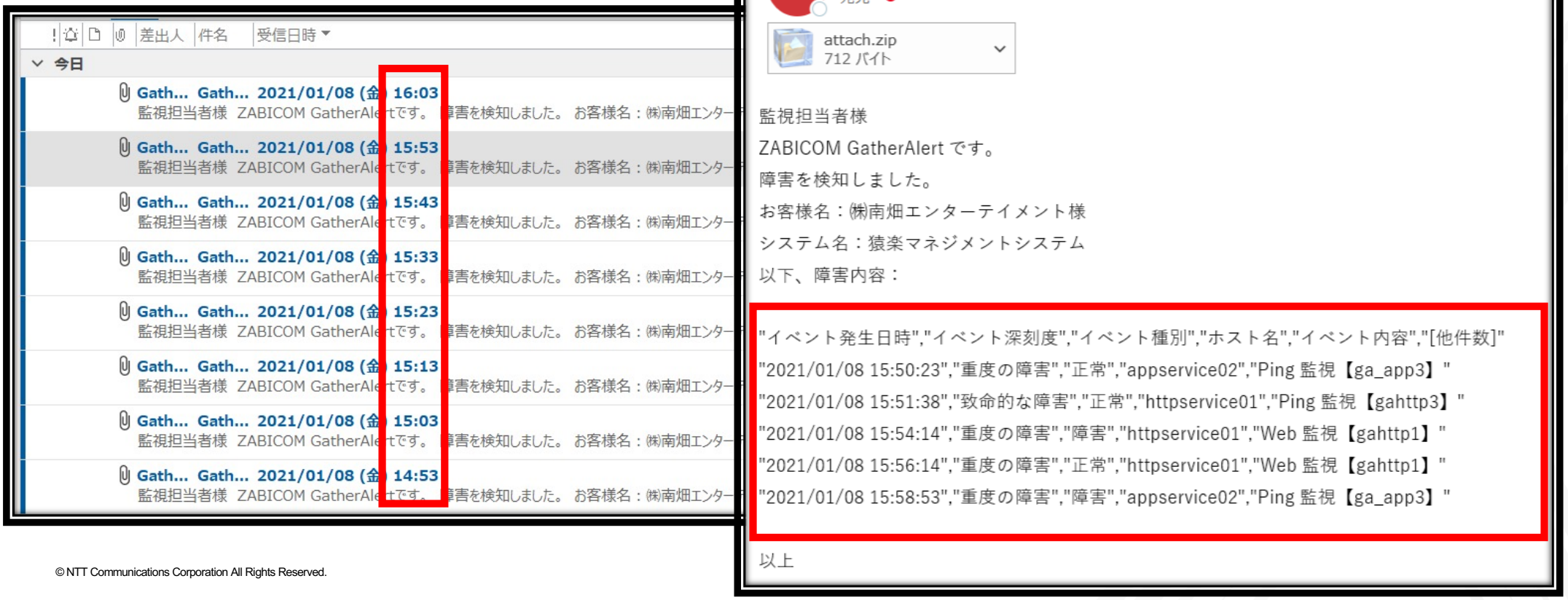
✓そもそも通知が多くて重要な障害を見落とす可能性もある

アラートの集約機能＆自動クローズ可能なツール

「**GatherAlert**」があります！

何ができるの？

一定期間おきに、障害内容をまとめて通知できる



何ができるの？

GatherAlert【重度・致命的障害】

GatherAlert@zabicom.com
宛先

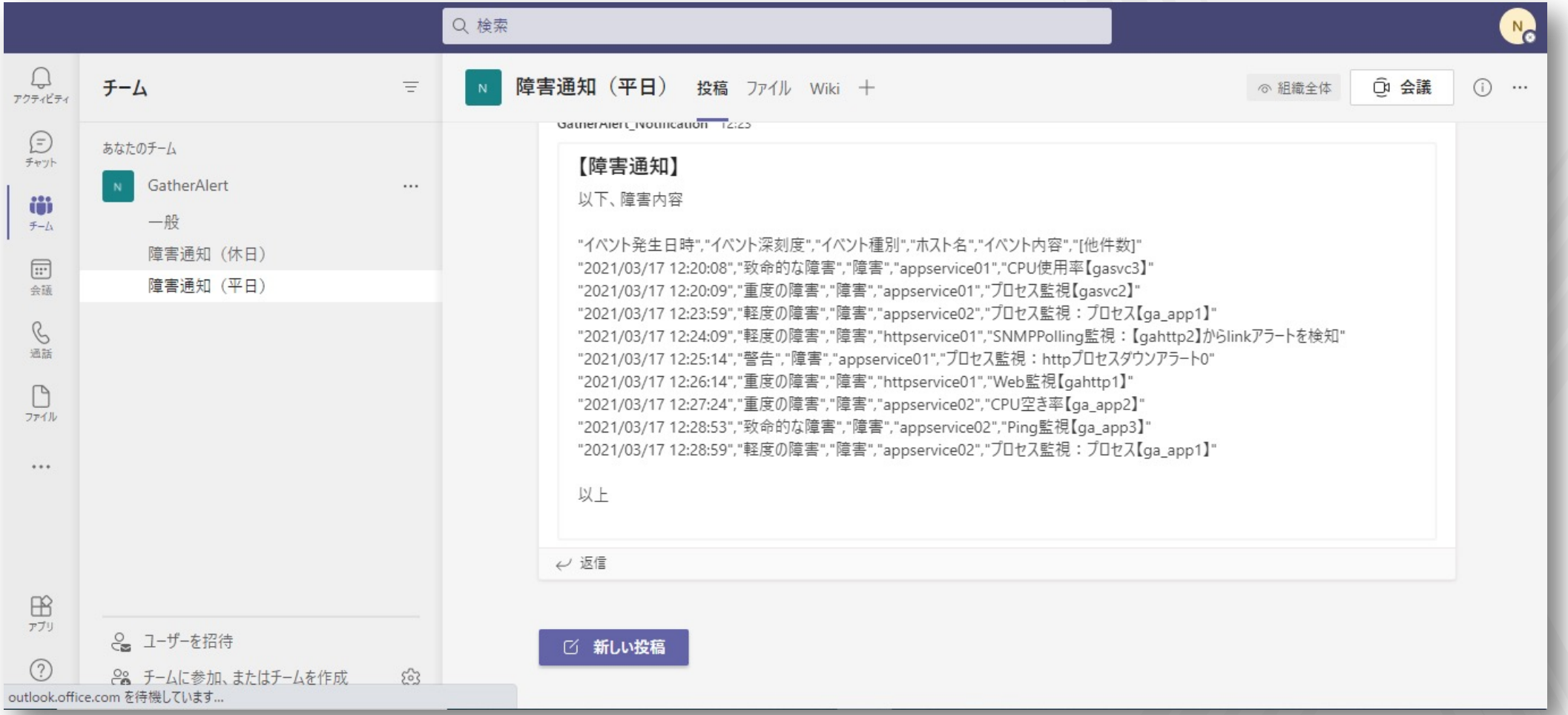
attach.zip
712 バイト

csv出力可能！

監視担当者様
ZABICOM GatherAlert です。
障害を検知しました。
お客様名：㈱南畑エンターテイメント様
システム名：猿楽マネジメントシステム
以下、障害内容：

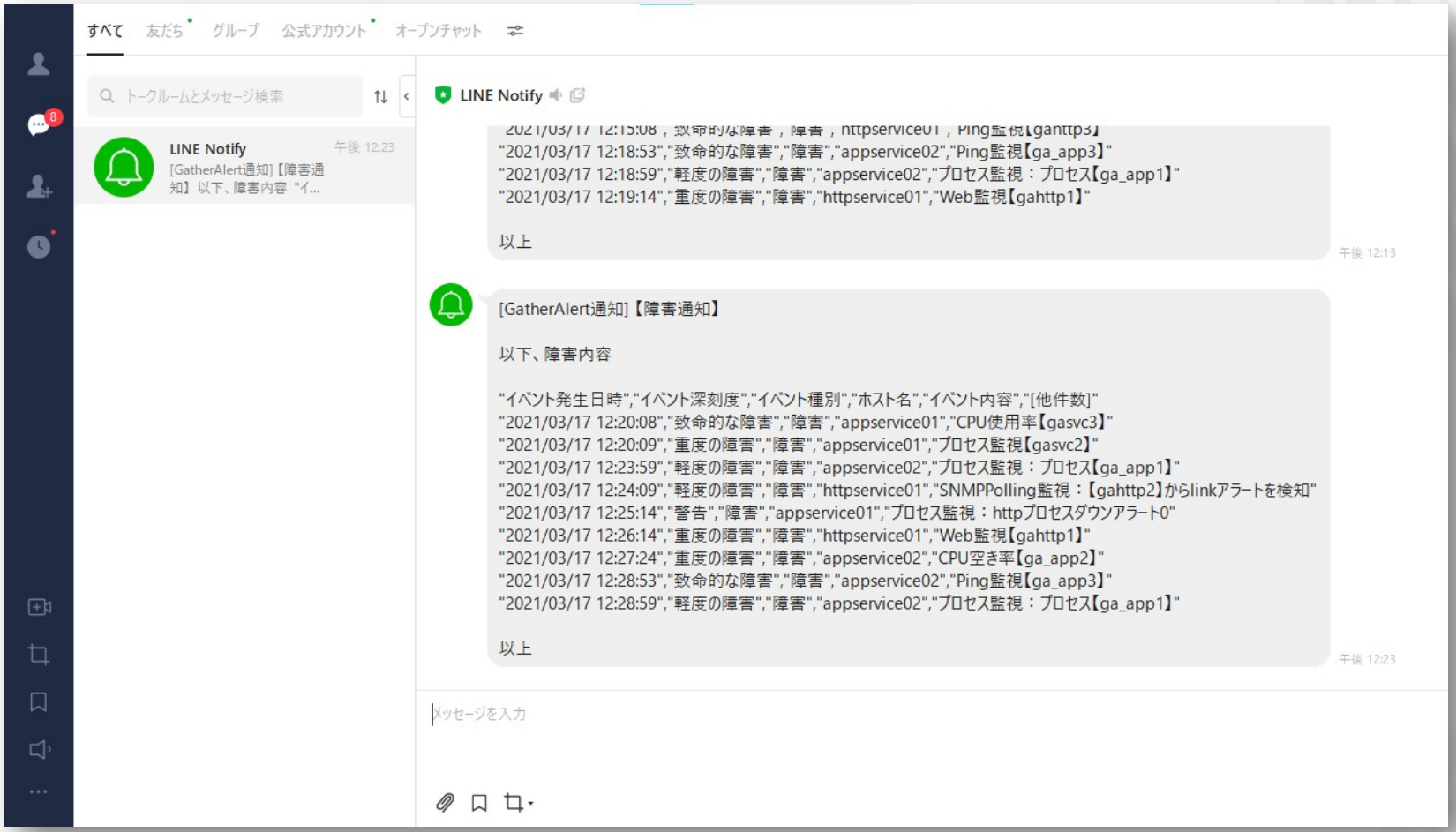
"イベント発生日時","イベント深刻度","イベント種別","ホスト名","イベント内容","[他件数]"
"2021/01/08 15:50:23","重度の障害","正常","appservice02","Ping 監視【ga_app3】"
"2021/01/08 15:51:38","致命的な障害","正常","httpservice01","Ping 監視【gahttp3】"

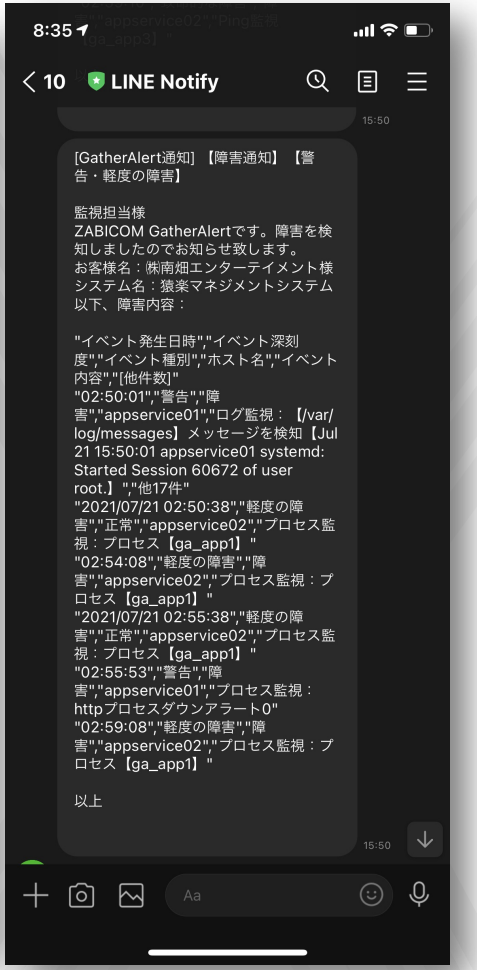
通知イメージ（Teams）



通知イメージ（Slack）

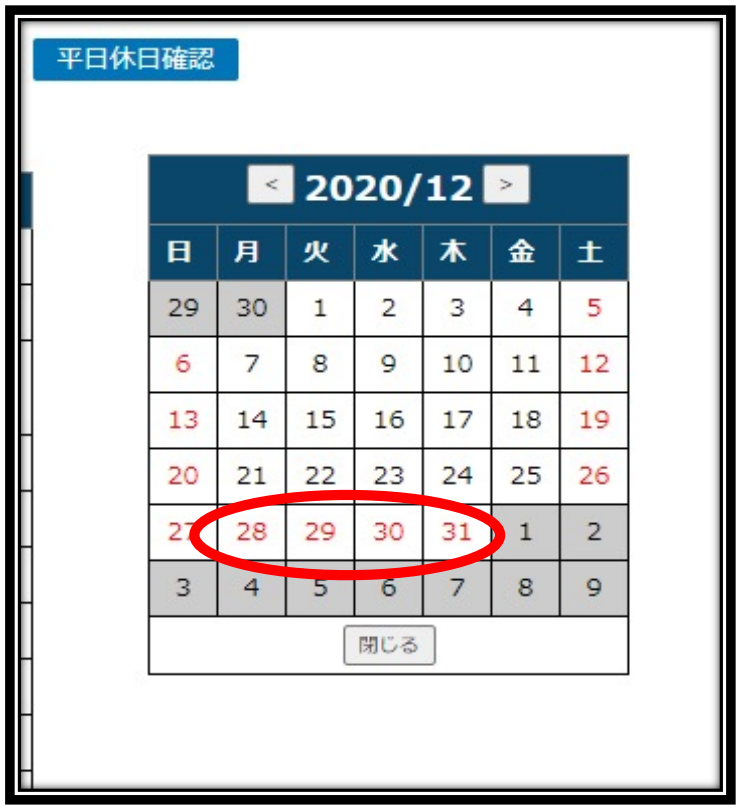
通知イメージ（LINE）





さらに

営業日と営業時間を自由に設定することができる



カレンダー機能
実装してますが何か？

どんなメリットがある？

1. アラートがまとまって通知される
2. 営業日・営業時間に応じて通知先を振り分けられる
3. 通知した障害については**自動でクローズ**する

まとめ

発表内容のまとめ

- ✓ 何でAll in Oneコンテナで実装したの？
- ✓ problemテーブル肥大化に関する事例と対応について
- ✓ GatherAlertの紹介

ご聴講ありがとうございました！