

ZABBIX 2021 Conference JAPAN



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

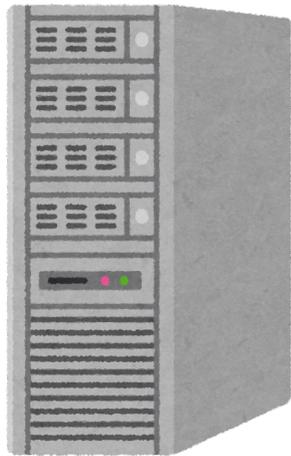
どんな要件があるのか？

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

今回お話しするケースでは…

検証環境でOKが出たものを、本番環境に適用したい

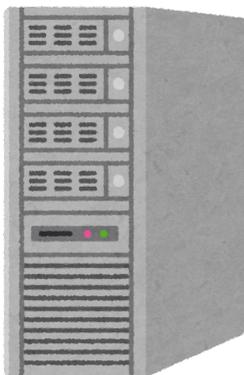
という要件も…



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



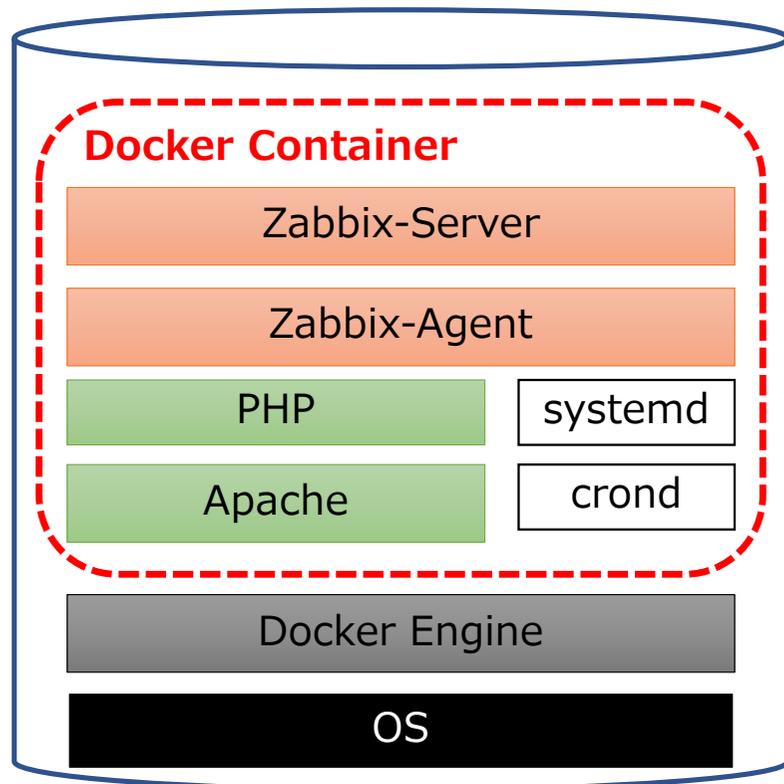
2～3時間

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)

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

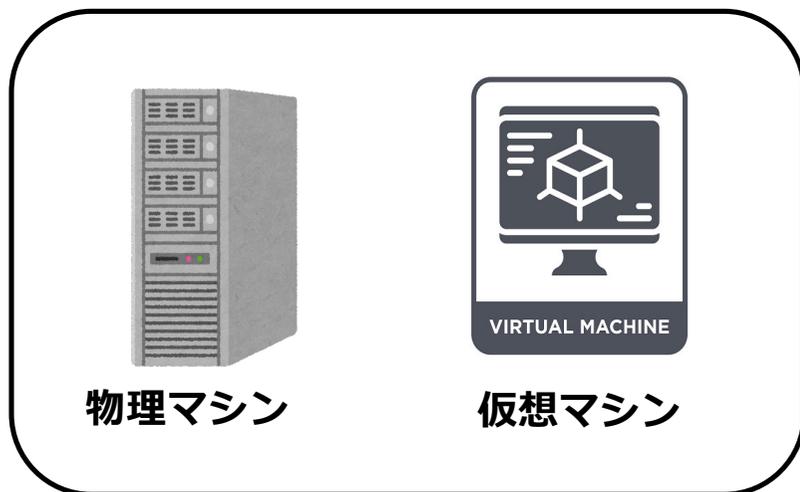
メリット

- ✓ 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…



運用実績の多さ

いいとこどり◎



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



The screenshot shows the ZABBIX interface for the '障害' (Incidents) page. The '表示' (Display) tab is highlighted with a red box, showing '最近の障害' (Recent Incidents) and '障害' (Incidents) options. The page includes search filters for host groups and hosts, and a 'CSVエクスポート' (Export to CSV) button.

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

The screenshot shows the Zabbix Global view dashboard. On the left is a navigation menu with options like '監視データ', 'ダッシュボード', '障害', 'ホスト', etc. The main content area is titled 'Global view' and contains a 'システム情報' (System Information) table and a '障害' (Problems) table. A bar chart titled '深刻度ごとの障害数' (Number of problems by severity) is highlighted with a red box. The bar chart shows the following counts: 0 for '致命的な障害' (Critical), 0 for '重度の障害' (High), 0 for '軽度の障害' (Medium), 3 for '警告' (Warning), 0 for '情報' (Information), and 0 for '未分類' (Unclassified). Above the bar chart, a summary row shows: 3 '利用可能' (Available), 0 '利用不可' (Unavailable), 0 '不明' (Unknown), and 3 '合計' (Total).

パラメータ	値	詳細
Zabbixサーバーの起動	(はい)	localhost:10051
ホスト数 (有効/無効)	20	3 / 17
テンプレート数	249	
アイテム数 (有効/無効/取得不可)	437	153 / 277 / 7
トリガー数 (有効/無効 [障害/正常])	382	126 / 254 [3 / 125]
ユーザー数 (オンライン)	2	1
1秒あたりの監視項目数(Zabbixサーバーの要求パフォーマンス)	6.04	

深刻度	数
致命的な障害	0
重度の障害	0
軽度の障害	0
警告	3
情報	0
未分類	0

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

トリガー

すべてのホスト / HostA 有効 ZBX SNMP JMX IPMI アプリケーション アイテム 20 トリガー 19 グラフ ディス

トリガー タグ 依存関係

*名前 ろく検知

運用データ

深深度 未分類 情報 警告 軽度の障害 重度の障害 致命的な障害

*条件式 {HostA:log[/hoge/moga.log].regexp(ERROR)}=1 追加

条件式ビルダー

正常イベントの生成 条件式 復旧条件式 なし

障害イベント生成モード 単一 複数

正常時のイベントクローズ すべての障害 タグの値が一致したすべての障害

手動クローズを許可

- ✓ 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



The screenshot shows the Zabbix web interface for configuring a trigger. The left sidebar contains navigation menus for Monitoring Data, Inventory, Reports, Settings, Host Groups, Templates, Hosts, Maintenance, Actions, Event Correlation, Discovery, and Services. The main content area is titled 'Trigger' and shows configuration for a trigger named 'ろく検知'. The trigger is active and associated with HostA. The severity is set to '警告' (Warning). The trigger condition is defined as `{HostA:log[/hoge/moga.log,ERROR].regexp(ERROR)}=1`. Below this, the 'Normal event generation' section is highlighted with a red box, showing the 'Recovery condition' tab selected with the condition `{HostA:log[/hoge/moga.log,ERROR].nodata(30)}=1`. The 'Event generation mode' is set to 'Single', and the 'Normal event close' is set to 'Close all problems'. At the bottom, the 'Allow manual close' checkbox is checked and highlighted with a red box. The URL field is empty.

復旧条件式の設定方法（regex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 $\{ \text{HostA:log}[\text{/var/log/messages,ERROR}].\text{regex}(\text{ERROR},\mathbf{30}) \} = 1$

アイテムキー

regex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 $\{ \text{HostA:log}[\text{/var/log/messages,ERROR}].\text{nodata}(30) \} = 1$

アイテム：監視データ収集の定義

トリガー：アイテムの障害判定の定義

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

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ます◎

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"

障害!

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



ZABBIX アイテム

すべてのホスト / appservice01 有効 ZBX SNMP JMX IPMI

アプリケーション 1 アイテム 18 トリガー 19 グラフ 1 ディスカバリールール Webシナリオ

アイテム 保存前処理

* 名前 エラーログ監視 【/hoge/moga.log】

タイプ Zabbixエージェント(アクティブ) ▼

* キー log[/hoge/moga.log_ERROR]

データ型 ログ ▼

**アイテムキーの
第2引数を指定！**

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

!	☆	□	📧	差出人	件名	受信日時
▼						今日
📧				Gath...	Gath...	2021/01/08 (金) 16:03
				監視担当者様	ZABICOM GatherAlert	です。障害を検知しました。お客様名：(株)南畑エンター
📧				Gath...	Gath...	2021/01/08 (金) 15:53
				監視担当者様	ZABICOM GatherAlert	です。障害を検知しました。お客様名：(株)南畑エンター
📧				Gath...	Gath...	2021/01/08 (金) 15:43
				監視担当者様	ZABICOM GatherAlert	です。障害を検知しました。お客様名：(株)南畑エンター
📧				Gath...	Gath...	2021/01/08 (金) 15:33
				監視担当者様	ZABICOM GatherAlert	です。障害を検知しました。お客様名：(株)南畑エンター
📧				Gath...	Gath...	2021/01/08 (金) 15:23
				監視担当者様	ZABICOM GatherAlert	です。障害を検知しました。お客様名：(株)南畑エンター
📧				Gath...	Gath...	2021/01/08 (金) 15:13
				監視担当者様	ZABICOM GatherAlert	です。障害を検知しました。お客様名：(株)南畑エンター
📧				Gath...	Gath...	2021/01/08 (金) 15:03
				監視担当者様	ZABICOM GatherAlert	です。障害を検知しました。お客様名：(株)南畑エンター
📧				Gath...	Gath...	2021/01/08 (金) 14:53
				監視担当者様	ZABICOM GatherAlert	です。障害を検知しました。お客様名：(株)南畑エンター

GatherAlert【重度・致命的障害】

 GatherAlert@zabicom.com
宛先

 attach.zip
712 バイト

監視担当者様
ZABICOM GatherAlert です。
障害を検知しました。
お客様名：(株)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】"
"2021/01/08 15:54:14","重度の障害","障害","httpservice01","Web 監視【gahttp1】"
"2021/01/08 15:56:14","重度の障害","正常","httpservice01","Web 監視【gahttp1】"
"2021/01/08 15:58:53","重度の障害","障害","appservice02","Ping 監視【ga_app3】"

以上

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：(株)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)



The screenshot shows the Microsoft Teams interface. On the left is a navigation pane with icons for 'アクティビティ', 'チャット', 'チーム', '会議', '通話', 'ファイル', and 'アプリ'. The main area displays a channel named '障害通知 (平日)' with a search bar at the top. A notification message is shown, titled '【障害通知】' with the subtitle '以下、障害内容'. The message content is a list of log entries for various services, including CPU usage, process monitoring, and ping monitoring. At the bottom of the message area is a '返信' (Reply) button. Below the message area is a '新しい投稿' (New Post) button. The bottom status bar shows 'outlook.office.com を待機しています...'.

検索

チーム

あなたのチーム

GatherAlert

一般

障害通知 (休日)

障害通知 (平日)

組織全体

会議

障害通知 (平日)

【障害通知】

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]"

"2021/03/17 12:20:08","致命的な障害","障害","appservice01","CPU使用率【gasvc3】"

"2021/03/17 12:20:09","重度の障害","障害","appservice01","プロセス監視【gasvc2】"

"2021/03/17 12:23:59","軽度の障害","障害","appservice02","プロセス監視：プロセス【ga_app1】"

"2021/03/17 12:24:09","軽度の障害","障害","httpservice01","SNMPPolling監視：【gahttp2】からlinkアラートを検知"

"2021/03/17 12:25:14","警告","障害","appservice01","プロセス監視：httpプロセスダウンアラート0"

"2021/03/17 12:26:14","重度の障害","障害","httpservice01","Web監視【gahttp1】"

"2021/03/17 12:27:24","重度の障害","障害","appservice02","CPU空き率【ga_app2】"

"2021/03/17 12:28:53","致命的な障害","障害","appservice02","Ping監視【ga_app3】"

"2021/03/17 12:28:59","軽度の障害","障害","appservice02","プロセス監視：プロセス【ga_app1】"

以上

返信

新しい投稿

ユーザーを招待

チームに参加、またはチームを作成

outlook.office.com を待機しています...

通知イメージ (Slack)



GatherAlert # 4t-a

+ Add a bookmark

以上

Wednesday, July 21st

GatherAlert_Notification APP 1:40 PM

【障害通知】 【重度・致命的障害】

監視担当様

ZABICOM GatherAlertです。障害を検知しましたのでお知らせ致します。

お客様名：(株)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]"

"2021/07/21 00:40:02";"重度の障害";"正常";"appservice01";"ログ監視";"他4件"

"00:40:02";"重度の障害";"障害";"appservice01";"ログ監視";"他4件"

"2021/07/21 00:40:40";"致命的な障害";"正常";"appservice02";"Ping監視【ga_app3】"

"00:40:54";"重度の障害";"障害";"appservice01";"プロセス監視【gasvc2】"

"00:42:39";"重度の障害";"障害";"appservice02";"CPU空き率【ga_app2】"

"2021/07/21 00:45:39";"重度の障害";"正常";"appservice02";"CPU空き率【ga_app2】"

"00:49:10";"致命的な障害";"障害";"appservice02";"Ping監視【ga_app3】"

以上

GatherAlert_Notification APP 1:50 PM

【障害通知】 【重度・致命的障害】

監視担当者様

ZABICOM GatherAlertです。障害を検知しましたのでお知らせ致します。

お客様名：(株)南畑エンターテイメント様

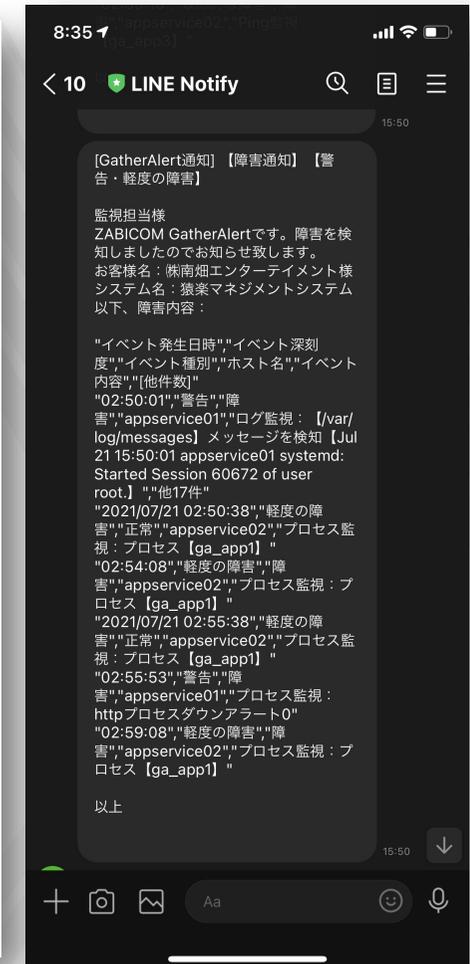
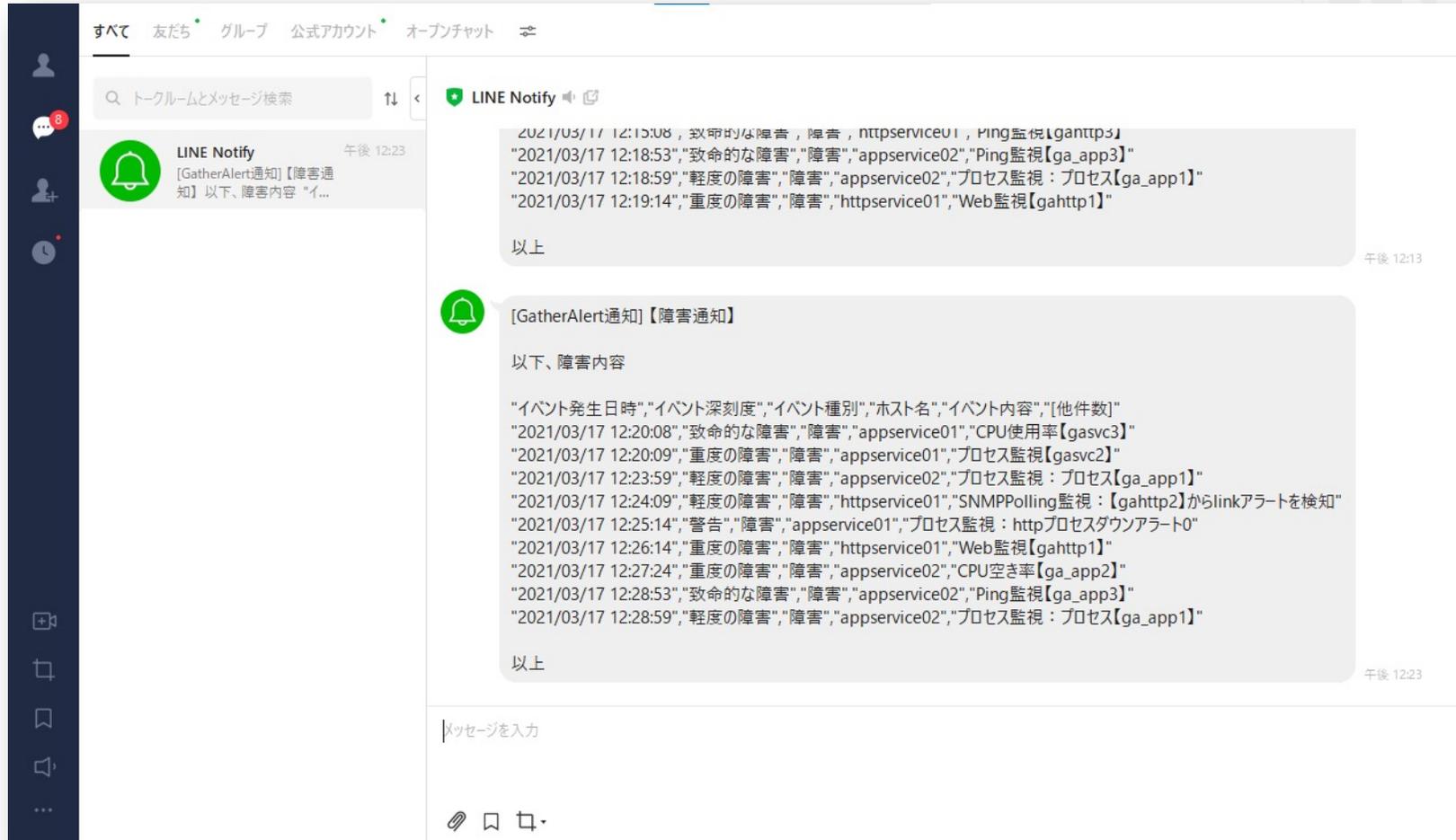
システム名：猿楽マネジメントシステム

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]"

Send a message to #4t-a

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



時間	深刻度	復旧時刻	ステータス	情報	ホスト	障害	継続期間	確認済	アクション
07:54:14	重度の障害	07:56:14	解決済		httpservice01	Web監視【qahttp1】...	2m	はい	🔊 3
07:53:59	軽度の障害	07:55:29	解決済		appservice02	プロセス監視：プロセス【ga_app1】...	1m 30s	はい	🔊 3
07:50:39	軽度の障害	07:52:39	解決済		httpservice01	SNMPPolling監視：【qahttp2】からlinkアラートを検知	2m	はい	🔊 3
07:50:38	致命的な障害	07:59:08	解決済		appservice01	CPU使用率【gasvc3】...	8m 30s	はい	🔊 3
07:49:38	致命的な障害	07:51:38	解決済		httpservice01	Ping監視【qahttp3】...	2m	はい	🔊 3
07:48:59	軽度の障害	07:50:29	解決済		appservice02	プロセス監視：プロセス【ga_app1】...	1m 30s	はい	🔊 3
07:48:53	致命的な障害	07:50:23	解決済		appservice02	Ping監視【ga_app3】...	1m 30s	はい	🔊 3
07:47:14	重度の障害	07:49:14	解決済		httpservice01	Web監視【qahttp1】...	2m	はい	🔊 3
07:43:59	軽度の障害	07:45:29	解決済		appservice02	プロセス監視：プロセス【ga_app1】...	1m 30s	はい	🔊 3
07:42:24	重度の障害	07:45:24	解決済		appservice02	CPU空き率【ga_app2】...	3m	はい	🔊 3

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！

