

Zabbixで 楽しんで監視を開始

2019年02月22日 @OSC 2019 TOKYO/SPRING

日本Zabbixユーザー会 田中敦



自己紹介

名前

- 田中 敦(たなか あつし)

所属

- 日本Zabbixユーザー会 副代表

主な活動

- コミュニティサイト(<http://www.zabbix.jp/>)でのQ&A対応
- ZabbixのWebインターフェースの日本語翻訳
- 雑誌やWebメディアでZabbixの紹介

@IT > クラウド > Linux & OSS > Zabbixは「自動化」で運用管理者の負担をどう減らし...

クラウド/コンテナ時代のZabbix再入門(前編) :

Zabbixは「自動化」で運用管理者の負担をどう減らしてきたのか(1/2)

Zabbixの概要をあらためて振り返り、Zabbixが「自動化」によって運用管理者の負担をどう減らしてきたのかを機能ごとに見ていく。

🕒 2018年12月11日 05時00分 公開

] 田中敦,



Zabbixとは

Zabbix (ザビックス) とは、オープンソースソフトウェア (OSS) の統合監視ツールです。開発元は、ヨーロッパの「ラトビア」という国にあるZabbix社。同社はZabbix本体のソースコードを全て公開しています。ライセンスはGPL v2です。

Zabbixは、ネットワークに接続されているサーバやネットワーク機器などの状態を把握して「正常に機能しているか」を監視し、問題があった場合、メールなどで運用管理者に通知したり、自動回復を試みたりします。

はじめに

はじめに

システムやサービスの監視をしないといけないとは思うけど、Zabbixのインストール面倒なんだよな、と思われている方々

Zabbixインストールしたはいいけど、監視対象の増減があると、その対応が面倒、と思われている方々

Zabbix 4.0とかリリースされて色々新機能があるらしいけど、何ができるかわからない、と思われている方々

ちょっと楽しんで
使い始めてみま
しょう

Zabbixの環境構築

普通にインストール

Linuxの基本的な知識があれば、RHEL(CentOS)、Ubuntu、Debian、Raspbian、SUSEなどの標準のパッケージを利用して環境を構築できる

CentOS 7上でZabbixサーバーを構築するのであれば、主要な手順は以下の手順だけ

1. Zabbixのリポジトリを追加
2. yumコマンドでインストール
3. データベースの作成と初期データの登録
4. データベース接続情報をZabbixサーバーの設定ファイルに記載

CentOS 7にZabbix 4.0をインストール(1)

```
# yum install http://repo.zabbix.com/zabbix/4.0/rhel/7/x86_64/zabbix-release-4.0-1.el7.noarch.rpm
# yum install zabbix-server-mysql zabbix-web-mysql zabbix-web-japanese zabbix-agent
# yum install mariadb-server
# vi /etc/my.cnf.d/server.cnf
-----
[mysqld]
character-set-server = utf8
collation-server    = utf8_bin
skip-character-set-client-handshake
innodb_file_per_table
-----
# systemctl start mariadb
# systemctl enable mariadb
# mysql -uroot
MariaDB [(none)]> create database zabbix character set utf8 collate utf8_bin;
MariaDB [(none)]> grant all privileges on zabbix.* to zabbix@localhost identified by 'password';
MariaDB [(none)]> exit
# zcat /usr/share/doc/zabbix-server-mysql-4.0.4/create.sql.gz | mysql -uroot zabbix
```

CentOS 7にZabbix 4.0をインストール(2)

```
# vi /etc/zabbix/zabbix_server.conf
-----
DBPassword=password
-----
# vi /etc/httpd/conf.d/zabbix.conf
-----
php_value date.timezone Asia/Tokyo
-----
# systemctl start zabbix-server
# systemctl start zabbix-agent
# systemctl start httpd
# systemctl enable zabbix-server
# systemctl enable zabbix-agent
# systemctl enable httpd
# firewall-cmd --add-port=10051/tcp --zone=public --permanent
# firewall-cmd --add-service=http --zone=public --permanent
# firewall-cmd --reload
```

あ、これが
めんどくさい？

Ansibleを使用して環境構築

Ansibleを使用すれば、有志の方が作成されたPlaybookを参考にして、そのPlaybookを実行するだけで環境構築が可能

AnsibleのZabbixモジュールも使用することでZabbixのAPIを活用した制御も可能

https://docs.ansible.com/ansible/latest/modules/list_of_monitoring_modules.html#zabbix

Ansibleを使って何ができるか

実現できることの例

- Zabbixエージェントのインストール
- Zabbixサーバーへのホスト登録
- ホストにインストールしたミドルウェアやアプリケーション用の監視テンプレートもリンクさせる

あ、これも
めんどくさい？

アプライアンスを使用

Zabbix社は公式にアプライアンスも公開していてダウンロードできる

- 提供されている形式
 - VirtualBox、VMware用 (.vmdk)
 - Hyper-V 2008/2012用 (.vhd、.vhdx)
 - KVM、Parallels、QEMU、Xen用 (.raw)
 - KVM、QEMU用 (.qcow2)
 - Open virtualization format (.ovf)

これを使用すれば、起動してネットワークの設定さえすればすぐに使える

アプライアンス環境の構築例

VirtualBox上にZabbix 4.0の環境を構築する

- アプライアンスのイメージをダウンロード
- VirtualBox上に空のVMを作成
- VMの設定で、ダウンロードしてきた圧縮ファイルから取り出したvmdxファイルをハードディスクとしてマウント
- VMを起動
- ネットワークインターフェースの設定を確認して修正
- 再起動して動作を確認

もうちょっと
新しめな方法も
ご紹介

Dockerコンテナを使用

Zabbix社は公式にDockerコンテナも公開している

個別にコンテナで起動することもできるが、Docker Composeを使用してまとめて起動することも可能

Zabbix公式のYAMLファイルそのままでは動かないので、少しだけ設定を追加して起動することでグラフの日本語フォントも合わせて設定することが可能

docker-composeでCentOSの環境準備

公式リポジトリ

<https://github.com/zabbix/zabbix-docker/>

MySQLを使用したCentOSベースの最新バージョン構築用のYAMLファイル

docker-compose_v3_centos_mysql_latest.yaml

docker-composeで環境起動(1)

GitHubからクローン

```
# git clone https://github.com/zabbix/zabbix-docker.git
```

いくつかの設定ファイル変更

```
.env_agent
```

```
-----
```

```
ZBX_HOSTNAME=Zabbix server
```

```
-----
```

```
.env_web
```

```
-----
```

```
PHP_TZ=Asia/Tokyo
```

```
-----
```

docker-composeで環境起動(2)

コンテナの起動

```
# docker-compose -f docker-compose_v3_centos_mysql_latest.yaml up -d
```

Webインターフェースでホストのエージェントのインターフェースを変更

127.0.0.1 IPアドレス → zabbix-agent DNS

docker-compose使用時の日本語フォント指定

公式のDockerイメージとYAMLではグラフの日本語フォント対応ができていない

zabbix/zabbix-web-apache-mysql:centos-4.0-latestにちょっと手を加えるため、Dockerファイルの用意とYAML内でbuild指定

Dockerfile :

```
FROM zabbix/zabbix-web-apache-mysql:centos-4.0-latest
RUN yum install -y vlgothic-p-fonts && ¥
  rm /usr/share/zabbix/fonts/graphfont.ttf && ¥
  ln -s /usr/share/fonts/vlgothic/VL-PGothic-Regular.ttf
  /usr/share/zabbix/fonts/graphfont.ttf
```

YAML変更箇所 :

```
.....
zabbix-web-apache-mysql:
  image: zabbix/zabbix-web-apache-mysql:centos-4.0-latest
  build: .
  ports:
```

Zabbixサーバー環境構築時注意点(1)

Zabbixは、監視するために取得した情報をデータベースに保存するので、特にデータベースのパフォーマンスと容量が重要

監視する対象の機器が多かったり、監視項目が多かったりして、1秒間に処理しなければならないデータ量が多い環境では、データベースサーバーとしては、メモリサイズが大きく、I/O性能の高いディスクが必要となる

コンテナを使用する場合は、永続的なデータを保存するために、ホストのディスクをコンテナにマウントして使うことになると思われるが、このディスクに関しても同様の配慮が必要

Zabbixサーバー環境構築時注意点(2)

Webインターフェースでグラフを表示したときに日本語が文字化けするのは、パッケージを使用して環境構築をしていれば、パッケージのインストールか日本語フォントをインストール後にパッケージの再設定をすれば解決する

- RHEL(CentOS)の場合

```
# yum install zabbix-web-japanese
```

- Ubuntu/Debianの場合

```
$ sudo apt-get install fonts-vlgothic
```

```
$ sudo dpkg-reconfigure zabbix-frontend-php
```

知識のある人へ、Zabbixの処理高速化

RHEL(CentOS)では、安定したソフトウェアを標準のパッケージに採用しているため、少し古いバージョンのソフトウェアで構成される

サードベンダーが提供しているパッケージと組み合わせて環境構築をできる知識があれば、最新バージョンのPHPを使用したり、Apache HTTP Serverの代わりにnginxを使用したりすることで、より高速化させることができる

Zabbix 4.0 + PHP 7.3 + nginxの環境構築

Zabbix社が公開しているパッケージは、多くの方が環境を容易に構築できるよう、各OSの標準パッケージを使用して構築できるようになっている

しかし、ディストリビューションによっては、使用しているパッケージのバージョンが古いため、様々なミドルウェアのバージョンアップによる改善の恩恵を得ることができない。

CentOS 7であっても、nginxのリポジトリなどを利用して極力パッケージを組み合わせより高速に処理できる環境を構築できる

パッケージ導入の事前準備

Zabbix4.0用のリポジトリ登録

```
# yum install http://repo.zabbix.com/zabbix/4.0/rhel/7/x86\_64/zabbix-release-4.0-1.el7.noarch.rpm
```

PHP 7.3用のリポジトリ登録

```
# yum install https://rpms.remirepo.net/enterprise/remi-release-7.rpm
```

PHP 7.3とZabbix 4.0のインストール

事前にPHPのパッケージをインストール済みであった場合は削除しておく

以下のコマンドラインを実行して一緒にPHPとZabbixをインストールする

```
(# yum remove php-common) ※PHPがインストール済みだった場合  
# yum --enablerepo=remi-php73 install zabbix-get zabbix-server-mysql ¥  
zabbix-web-mysql zabbix-web-japanese zabbix-agent
```

nginxのインストール

リポジトリの登録

```
# vi /etc/yum.repos.d/nginx.repo
```

```
[nginx-stable]
name=nginx stable repo
baseurl=http://nginx.org/packages/centos/$releasever/$basearch/
gpgcheck=1
enabled=1
gpgkey=https://nginx.org/keys/nginx_signing.key

[nginx-mainline]
name=nginx mainline repo
baseurl=http://nginx.org/packages/mainline/centos/$releasever/$basearch/
gpgcheck=1
enabled=0
gpgkey=https://nginx.org/keys/nginx_signing.key
```

インストールの実行

```
# yum install nginx
```

```
# yum --enablerepo=remi-php73 install php-fpm
```

nginxの設定(1)

php-fpmを使用するため、Zabbix用のPHP設定を行う

/etc/php.d/50-zabbix.ini :

```
max_execution_time = 300
memory_limit = 128M
post_max_size = 16M
upload_max_filesize = 2M
max_input_time = 300
date.timezone = "Asia/Tokyo"
```

起動ユーザーの変更

/etc/nginx/nginx.conf :

```
user apache;
```

nginxの設定(2)

不要な設定ファイルを読み込まないようにファイル名変更

```
# mv /etc/nginx/conf.d/default.conf /etc/nginx/conf.d/default.conf.bak
```

Zabbixのコンテンツ設定

```
# ln -s /usr/share/zabbix /usr/share/nginx/html/zabbix
```

nginxの設定(3)

/etc/nginx/conf.d/zabbix.conf :

```
server {
    listen 80;
    root /usr/share/nginx/html;
    index index.html;
    access_log /var/log/nginx/zabbix-access_log main;
    error_log /var/log/nginx/zabbix-error_log error;

    location /zabbix {
        index index.php;
    }

    location ~ \.php$ {
        fastcgi_pass 127.0.0.1:9000;
        fastcgi_index index.php;
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
        include fastcgi_params;
    }
}
```

/etc/nginx/fastcgi_params :

```
fastcgi_intercept_errors    on;
fastcgi_ignore_client_abort off;
fastcgi_connect_timeout    60;
fastcgi_send_timeout       180;
fastcgi_read_timeout       180;
fastcgi_buffer_size        128k;
fastcgi_buffers            4 256k;
fastcgi_busy_buffers_size  256k;
fastcgi_temp_file_write_size 256k;
```

さらにHTTP/2を使用

さらに、使用するプロトコルとしてHTTP/1.1ではなく、HTTP/2を使用することで、各描画オブジェクトのリクエスト処理を並列化でき、処理を並列化することで、Webページ全体の描画終了までの時間を短縮することができる

HTTP/2を使用するためには、HTTPからHTTPSへの変更も必要

ここでは、オレオレ証明書を使用して構築例を示す

オレオレ証明書を作成と配置

```
# yum install openssl openssl-devel openssl-libs
# mkdir /etc/nginx/ssl
# openssl req -new -x509 -sha256 -newkey rsa:2048 -days 365 -nodes -out
/etc/nginx/ssl/nginx.pem -keyout /etc/nginx/ssl/nginx.key
# chown root:root -R /etc/nginx/ssl/
# chmod 600 /etc/nginx/ssl/*
# chmod 700 /etc/nginx/ssl
```

HTTP/2の設定

/etc/nginx/conf.d/zabbix.conf :

```
server {
    listen 80;
    return 301 https://$host$request_uri;
}
server {
    listen 443 ssl http2;
    server_name _;
    ssl_certificate /etc/nginx/ssl/nginx.pem;
    ssl_certificate_key /etc/nginx/ssl/nginx.key;
    root /usr/share/nginx/html;
    index index.html;
    access_log /var/log/nginx/zabbix-access_log main;
    error_log /var/log/nginx/zabbix-error_log error;

    location /zabbix {
        index index.php;
    }

    location ~ \.php$ {
        fastcgi_pass 127.0.0.1:9000;
        fastcgi_index index.php;
        fastcgi_param SCRIPT_FILENAME
        $document_root$fastcgi_script_name;
        include fastcgi_params;
    }
}
```

監視設定の効率化

自動化機能の活用

Zabbixには、監視設定を自動化、もしくは省力化するために以下のような機能が用意されている

- テンプレート
- ネットワークディスカバリ
- 自動登録
- LLD(Low Level Discovery)
- API

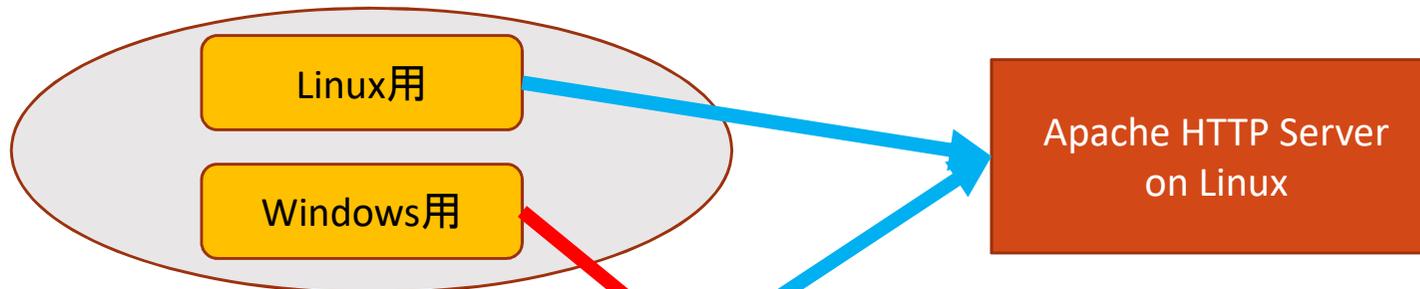
テンプレートの活用

テンプレートは、監視項目(アイテム)、障害条件の設定(トリガー)、グラフ、イベント発生状況やグラフなどをまとめた画面(スクリーン)をまとめたもの

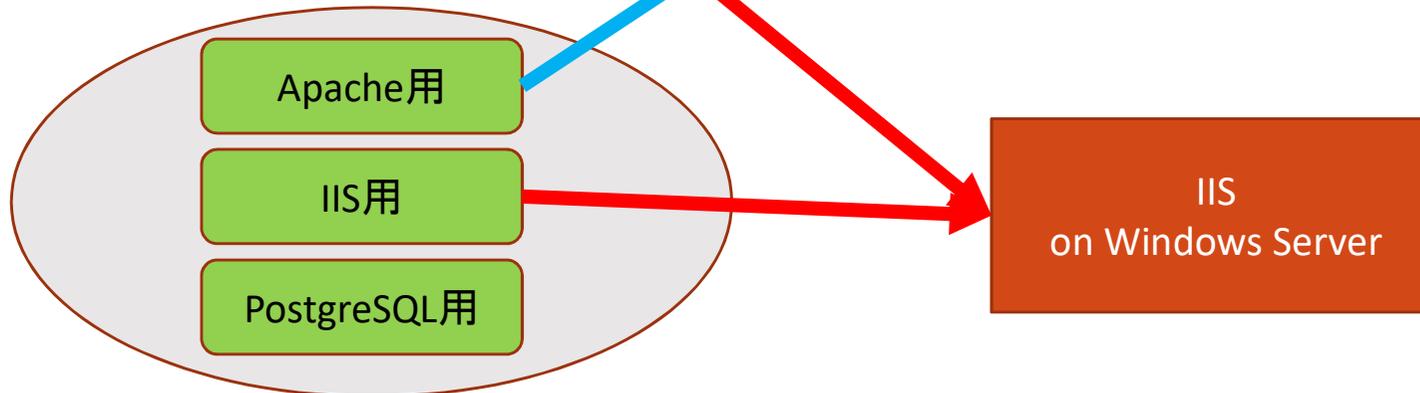
OSやミドルウェア、アプリケーションごとに用意しておくことで、各ホストの役割に応じた監視を簡単に開始できる

テンプレートの構成法

各種OS用テンプレート



各種ミドルウェア用テンプレート



ネットワークディスカバリの活用

IPアドレスのレンジを指定して、そのレンジ内のIPアドレス内を探索して、指定しておいた条件の応答が得られたら、デバイスのディスカバリイベントを発生させる

ホストとして登録するには、合わせてディスカバリのイベントを受け取った時のアクションも作成することが必要

あまり広いIPアドレスのレンジを指定するとそのレンジ内の探索が終了するまでに時間がかかってしまうので、クラスC程度にしておくこと

自動登録機能の活用

Zabbixエージェントを起動してZabbixサーバーに接続したときに、Zabbixサーバーに登録されていなかったら、ホストとして登録する機能

事前にZabbixサーバーに監視対象のホスト登録の作業が不要となるため、監視対象の追加の自動化に有効

ホスト登録時に、HostMetadataというZabbixエージェントの設定内の項目を使用して、その文字列に含まれる文字列などでリンクさせるテンプレートも指定可能

LLDの活用

標準で用意されているのは、ホストに接続されているネットワークインターフェースや、ホストにマウントされているハードディスクを自動的に認識して、それぞれのインターフェースやハードディスク用の監視項目を設定できる

独自のLLDを定義することも可能

- 例
 - AWSの複数のインスタンスを自動的に探索
 - DBMS上に定義されている複数のDBを自動的に探索
 - ホスト上に登録されているDockerコンテナを自動的に探索

LLDでの閾値の変更

マウントされた複数のハードディスクがあった時に、障害とみなすトリガーの閾値が「空き容量が20%」となっていると、ハードディスクのサイズが大きなディスクだと、まだまだ数十GB空きがあるのに障害とみなされてしまう場合がある

マウントポイントごとに閾値を指定できるように、トリガーの条件式で指定する閾値にマクロを使用する

例: /(ルート)なら10%、/mnt/data なら5%など

https://www.zabbix.com/documentation/4.0/manual/discovery/low_level_discovery#using_lld_macros_in_user_macro_contexts

APIの活用

APIを使用することで、他のシステムやツールからZabbixを制御できる

- Zabbixサーバー上の設定確認と変更
- Zabbixサーバーが収集したアイテムのデータや発生した障害イベントの情報を取得

動的に変化する環境の 監視

各種環境の監視設定共有サイト利用

Zabbix Shareに登録されているテンプレートを活用

Zabbix Share

<https://share.zabbix.com/>

公開されているものの一部

- AWS
- Azure
- KVM、Hyper-V、Vmware
- Docker

Docker

2015年10月にOSCで私が紹介した監視方法は、cgroupの各ステータスファイルを参照する方法

ネットワークのI/Oも見るためにはZabbixエージェントをrootユーザーで起動することが必要

rootユーザーを使用しなくても、cAdvisorを利用することでネットワークI/Oを含む各種メトリクスを取得可

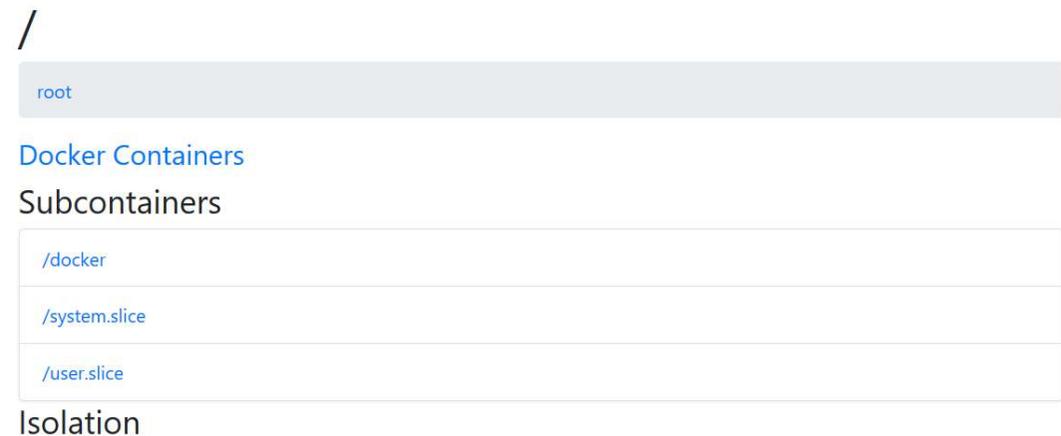
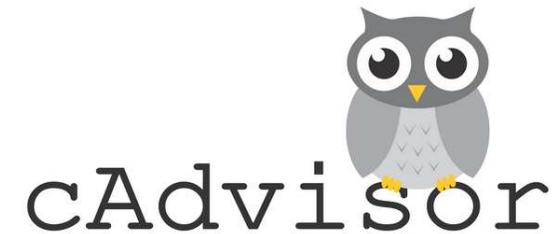
cAdvisorを使用したコンテナ環境のメトリクス取得

cAdvisor

<https://github.com/google/cadvisor>

cAdvisorを起動すると、同じホナのメトリクスを取得すること

cAdvisorのAPIを使用してZabbixで監視



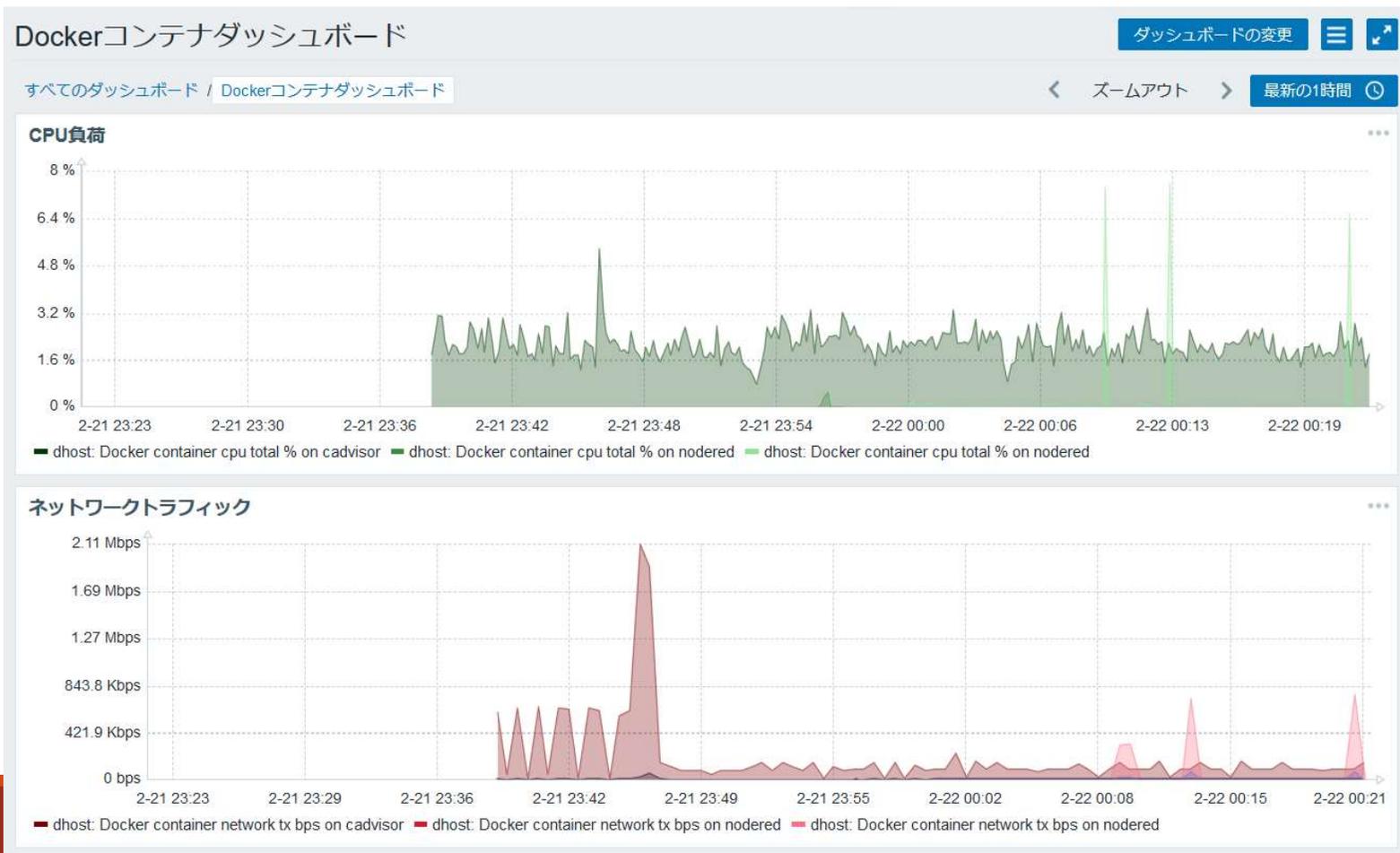
cAdvisorとZabbixとの連携

用意するスクリプト

- コンテナディスカバリ
- 各コンテナのメトリクス取得
 - CPU
 - メモリ
 - ネットワークI/O

これらを組み合わせることで起動されたDockerコンテナの増減を動的に監視可能

Docker環境監視例



各コンテナのCPU負荷
状況(コンテナの動的
増減に対応)

各コンテナのネット
ワーク送受信トラフィッ
ク(コンテナの動的増
減に対応)

Zabbixの今後

次のZabbix 4.2に実装されそうな機能

Beta版で実装済

- 保存前処理で値のバリデーションを実施
- 保存前処理でJavaScriptを使用して処理を高度化
- バックエンドのデータベースとしてTimescaleDBにも対応

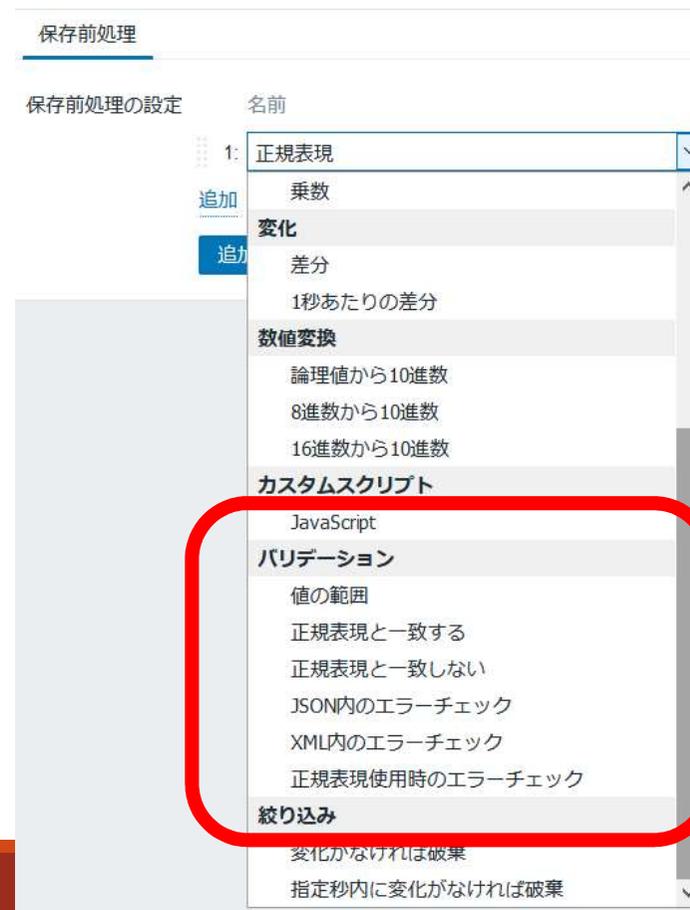
ロードマップにあるもの

- アクションやメディアのテスト機能

保存前処理でのバリデーション

保存前処理で値チェック

- 値の範囲
- 正規表現との比較
- JSON
- XML



保存前処理でJavaScript使用

JavaScriptで独自のロジックを記述して値を加工し、その処理結果を次の保存前処理に渡したり、アイテムの値とすることが可能

設定サンプル: Apacheのステータス取得

ZBXNEXT-4944 : Javascript for preprocessing

<https://support.zabbix.com/browse/ZBXNEXT-4944>

TimescaleDBへの実験的な対応

TimescaleDB

<https://www.timescale.com/>

特徴

- PostgreSQLをベースに時間ベースの自動的なパーティショニングに対応し、高速でスケーラブルなソリューション

設定方法

- Migration to TimescaleDB

<https://www.zabbix.com/documentation/4.2/manual/appendix/install/timescaledb>

さらに予定されている機能追加と改善

詳細は、「Zabbix Roadmap」を参照

- Zabbix Roadmap

<https://www.zabbix.com/roadmap>

まとめ

最後に

Zabbixで用意されている機能を活用して、動的な環境であっても監視の設定を自動化することで、運用管理者への負荷を軽減できる

今後の機能拡張でも、様々な設定の省力化や設定確認方法の実装など、設定作業を中心として作業負荷の軽減が期待される

不明なことがあれば、コミュニティも活用を

ご清聴いただき
ありがとうございました。

本資料内で利用させて頂いた各プロダクト名や
サービス名などは、各社もしくは各団体の商標ま
たは登録商標です。

おまけ

書籍発売決定

[改訂2版]Zabbix統合監視徹底活用
複雑化・大規模化するインフラの一元管理

著者:池田大輔

価格:本体3,600円+税

ISBN:978-4-297-10434-4

2019年2月27日発売予定

Software Design plus

改訂2版 ザビックス **Zabbix** [Version 4.0対応]
統合監視

複雑化・大規模化する
インフラの一元管理

TIS株式会社
池田大輔 [著]

徹底活用

オンプレミス・クラウド・コンテナを
シンプルかつ柔軟に自動監視する

- SNMP、IPMIを活用した物理環境の監視
- vSphere環境、Docker環境の監視
- AWSのアクセス状況、課金情報の監視
- 暗号化通信対応によるセキュアな環境の構築
- ダッシュボード機能やGrafana連携による可視化
- ElasticsearchやFluentd連携によるログ解析
- アクション機能による障害への自動対応
- Ansible連携による環境構築の自動化

技術評論社

参考URL

Zabbix 4.0のご紹介

<https://www.slideshare.net/atanaka7/zabbix-40-osc-2018-tokyofall-120981667>