

ZabbixでDockerも 監視してみよう

2015/10/24 @OSC 2015 TOKYO/FALL

ZABBIX-JP 田中 敦

自己紹介

▶ 名前

▶ 田中 敦 (たなか あつし)

▶ Twitter: @atanaka7

▶ 所属

▶ ZABBIX-JP (統合監視ツールZabbixの日本のコミュニティ) 副代表

▶ 普段は某Sierのエンジニア

▶ 生息地

▶ ZABBIX-JPのサイト (<http://www.zabbix.jp/>)

今日のアジェンダ

- ▶ Zabbixとは？
- ▶ 動的に変化するインフラ環境
- ▶ 監視項目増大による負荷上昇
- ▶ Dockerとは？
- ▶ ZabbixでDocker環境を監視
- ▶ Docker環境監視の実装例
- ▶ まとめ

Zabbixとは？

サーバーやネットワークの監視なら

◎みんな大好き(大嫌い?)Zabbix!

Zabbixとは

- ▶ オープンソースの統合監視ツール
- ▶ ヨーロッパにあるラトビアという国でZabbix SIAという会社が開発
- ▶ 初期バージョンがリリースされて約10年経過
- ▶ 監視用エージェントを各サーバにインストールして監視
- ▶ UNIX、Linux、Windowsなどの各OSに対応したエージェントあり
- ▶ エージェントがなくても、SNMP、ssh、telnet、IPMI、JMXなどでも監視可
- ▶ 監視の設定はWebブラウザからできる
- ▶ 監視設定にはテンプレートという監視設定パターンをまとめておくと、新しい監視対象のサーバが追加されたら紐づけするだけで監視開始
- ▶ 監視のために取得した各種データは、グラフ化して傾向を把握しやすい
- ▶ 障害検知時に、メールやメッセージを送るだけではなく、独自のスクリプトを実行可能
- ▶ マルチテナントの機能を利用して、同じZabbixサーバ上で利用者を分離可能
- ▶ などなどなどなどなどなどなど...

Zabbixの画面イメージ

7

The image displays a screenshot of the Zabbix 2.4.6 web interface, showing several panels and graphs. The interface is in Japanese and includes a navigation menu, a dashboard, and various monitoring tools.

Dashboard (ダッシュボード): Shows the Zabbix logo and navigation options like '監視データ' (Monitoring Data), 'インベントリ' (Inventory), and 'レポート' (Reports). It also features a sidebar with 'お気に入りのグラフ' (Favorite Graphs), 'お気に入りのスクリーン' (Favorite Screens), and 'お気に入りのマップ' (Favorite Maps).

ネットワークマップ (Network Map): Displays a map of Japan with a highlighted region labeled '福岡 正常' (Fukuoka Normal).

トリガーのステータス (Trigger Status): A table showing the status of 26 triggers. The table has columns for '選別度' (Priority), 'ステータス' (Status), and '情報' (Info). The status column shows various levels of alerts (警告, 正常, 障害) and their corresponding colors.

カスタムグラフ (Custom Graphs): Two panels showing custom graphs for 'CPU load' and 'Disk space usage /'. The CPU load graph shows a line chart with a legend for 'Processor load (1 min average)', 'Processor load (5 min average)', and 'Processor load (15 min average)'. The Disk space usage graph shows a pie chart for 'tkyweb01: Disk space usage / (1h)' with a legend for 'Total disk space on / [平均]' (17.11 GB, 100%) and 'Free disk space on / [平均]' (15 GB, 87.70%).

Footer: The footer contains the text 'Zabbix 2.4.6 Copyright 2001-2015 by Zabbix SIA' and 'ユーザー 'Admin' でログイン' (Logged in as user 'Admin').

動的に変化するインフラ環境

クラウドやコンテナ技術の発展

- ▶ 様々なクラウドやコンテナ技術が利用しやすくなり、活用されることによって、様々なサービスや機能を実現するために、色々な役割のサーバーを短期間で構築してサービスを開始することができるようになった。
- ▶ 以前は、物理的なサーバーの発注から納品までの時間が1週間とか数か月とかかかっていたのが、クラウドを利用すれば、Webブラウザ上から数分で利用開始ができる。

動的に増減する監視対象

- ▶ クラウドを利用することによって、監視しなければならないサーバーの数が増減する
- ▶ さらにDockerなどのコンテナ技術を活用しはじめると、各コンテナの状態の把握もしなければならない
- ▶ これらの監視対象が動的に増減するような環境を監視しなければならない

監視設定面倒！

- ☹ え～、またサーバー追加したの？
 - ☹ コンテナごとに色々監視設定するの面倒じゃん。
 - ☹ サーバーやコンテナを追加するたび監視設定の操作しないといけないの？
 - ☹ サーバーやコンテナを削除したときはどうするの？
-
- ◎ そんなあなたに「自動登録」と「ローレベルディスクバリ」

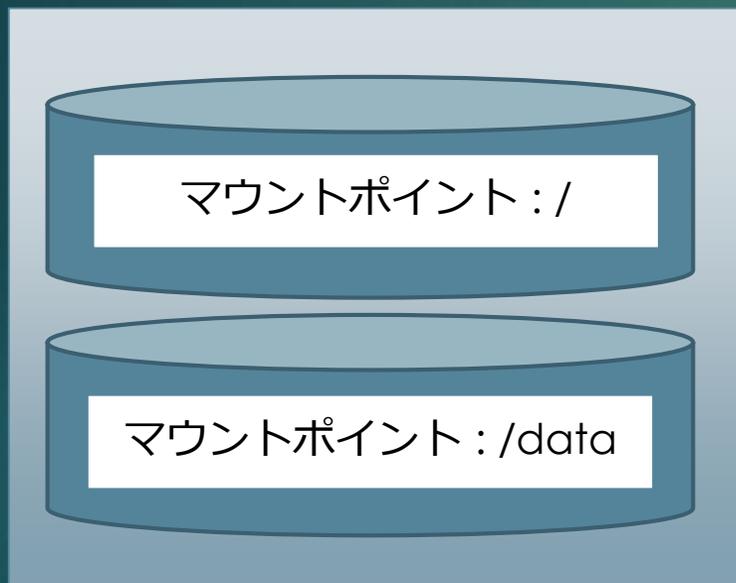
自動登録とは

- ▶ サーバーを追加して、それを監視対象としてZabbixサーバー上に自動登録する方法に関しては、OSC 2015 Tokyo/Springで発表した「全自動Zabbix」を参照してください。
 - ▶ 「全自動Zabbix 2.2&2.4」
<http://www.slideshare.net/qryuu/zabbix2224>
- ▶ Zabbixサーバー上に「自動登録」の設定を行って、Zabbixエージェントの設定を工夫すれば、Zabbixエージェントを起動後に、自動的に監視対象のサーバーとしてZabbixサーバー上に登録して、それぞれのサーバーの役割に合わせた監視を開始することができます。

ローレベルディスクカバリとは

- ▶ Zabbix 2.0で実装された機能
- ▶ JSONフォーマットで監視対象のリストを返すものを用意すれば、それぞれに対して監視項目を自動生成可
- ▶ サーバ上のネットワークインターフェースやマウントされたディスクの台数が異なる場合でも、共通の監視設定で監視できるようにすることが元々の機能
- ▶ この機能を応用することで、動的に対象の数が増減するものであっても、自動的に監視設定の追加や削除をすることが可能

(例) ローレベルディスクカバリがないと

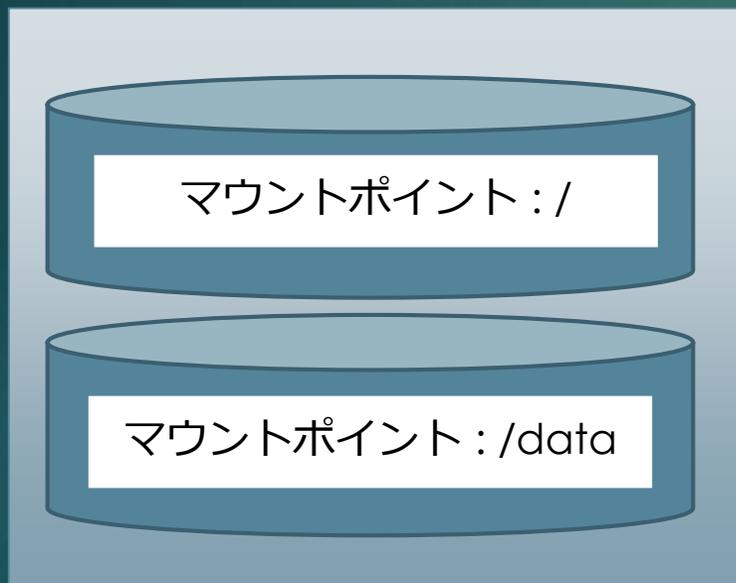


- ①サーバーにハードディスクを追加(手動)
- ②OSにハードディスクを認識させマウントする(手動)
- ③マウントしたハードディスク用の監視項目を作成する(手動)
- ④作成した監視項目に対する閾値を設定する(手動)
- ⑤作成した監視項目に対するグラフを作成する(手動)

サーバーごとに固有のマウントポイントであったりサイズが異なったりする場合があるので、汎用的なテンプレートとして事前準備できない。

(例)ローレベルディスクカバリがあると

15

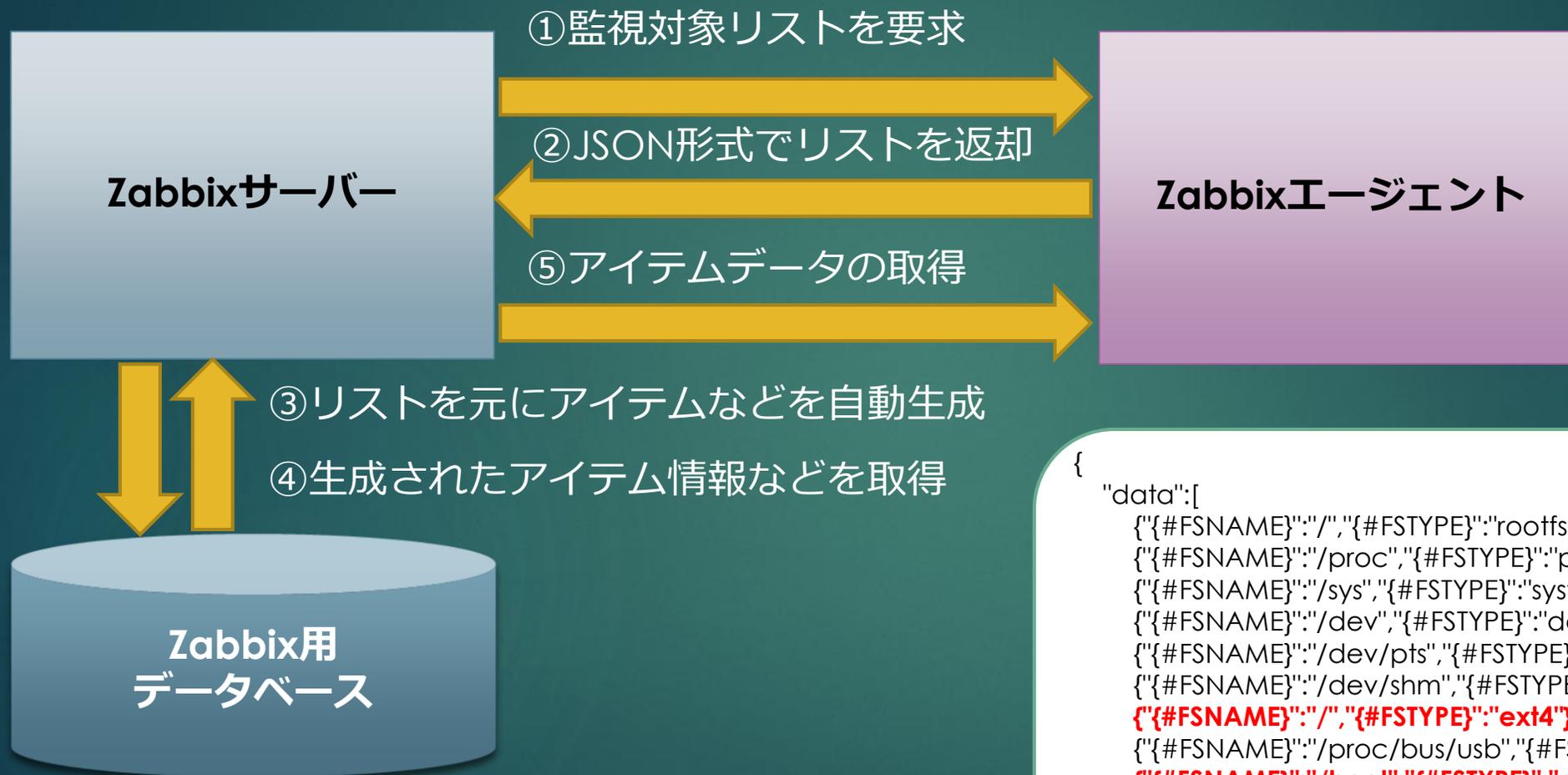


事前準備：サーバーにディスクバリルールを含むテンプレートを適用しておく

- ①サーバーにハードディスクを追加(手動)
- ②OSにハードディスクを認識させマウントする(手動)
- ③マウントされたハードディスクをZabbixエージェントが認識して、事前に設定しておいたディスクカバリのルールによって、追加されたディスクに対応した監視項目、閾値(トリガー)、グラフが自動生成されて、監視が開始される(自動)

事前に用意してあったテンプレート上のディスクカバリのルールが自動で適用されることで、マウントされたディスクの増減があっても手動での設定変更が不要。

ローレベルディスクカバリの実装



```
{
  "data": [
    {"#FSNAME": "/", "#FSTYPE": "rootfs"},
    {"#FSNAME": "/proc", "#FSTYPE": "proc"},
    {"#FSNAME": "/sys", "#FSTYPE": "sysfs"},
    {"#FSNAME": "/dev", "#FSTYPE": "devtmpfs"},
    {"#FSNAME": "/dev/pts", "#FSTYPE": "devpts"},
    {"#FSNAME": "/dev/shm", "#FSTYPE": "tmpfs"},
    {"#FSNAME": "/", "#FSTYPE": "ext4"},
    {"#FSNAME": "/proc/bus/usb", "#FSTYPE": "usbfs"},
    {"#FSNAME": "/boot", "#FSTYPE": "ext4"},
    {"#FSNAME": "/proc/sys/fs/binfmt_misc", "#FSTYPE": "binfmt_misc"}
  ]
}
```

監視項目増大による負荷上昇

動的に監視対象を抽出できても

- ▶ 自動登録やローレベルディスクバリの機能を利用することで、サーバーやコンテナの数が増減しても、自動的に監視設定を追加したり削除したりすることができる
- ▶ しかし、監視対象の数が増加してきた場合、Zabbixの標準機能で取得できないサーバーやコンテナの状態情報(メトリクス)が多いと、それらの値を取得するための処理の負荷が上昇する
- ▶ 例えば、スクリプトを実行して値を取得するような処理が多い場合、そのスクリプトを実行するためにプロセスを生成して、シェルを動かして、さらにその上でスクリプトの処理をして、となるので、数が増えるとサーバーへの負荷が高い

パフォーマンス悪そう

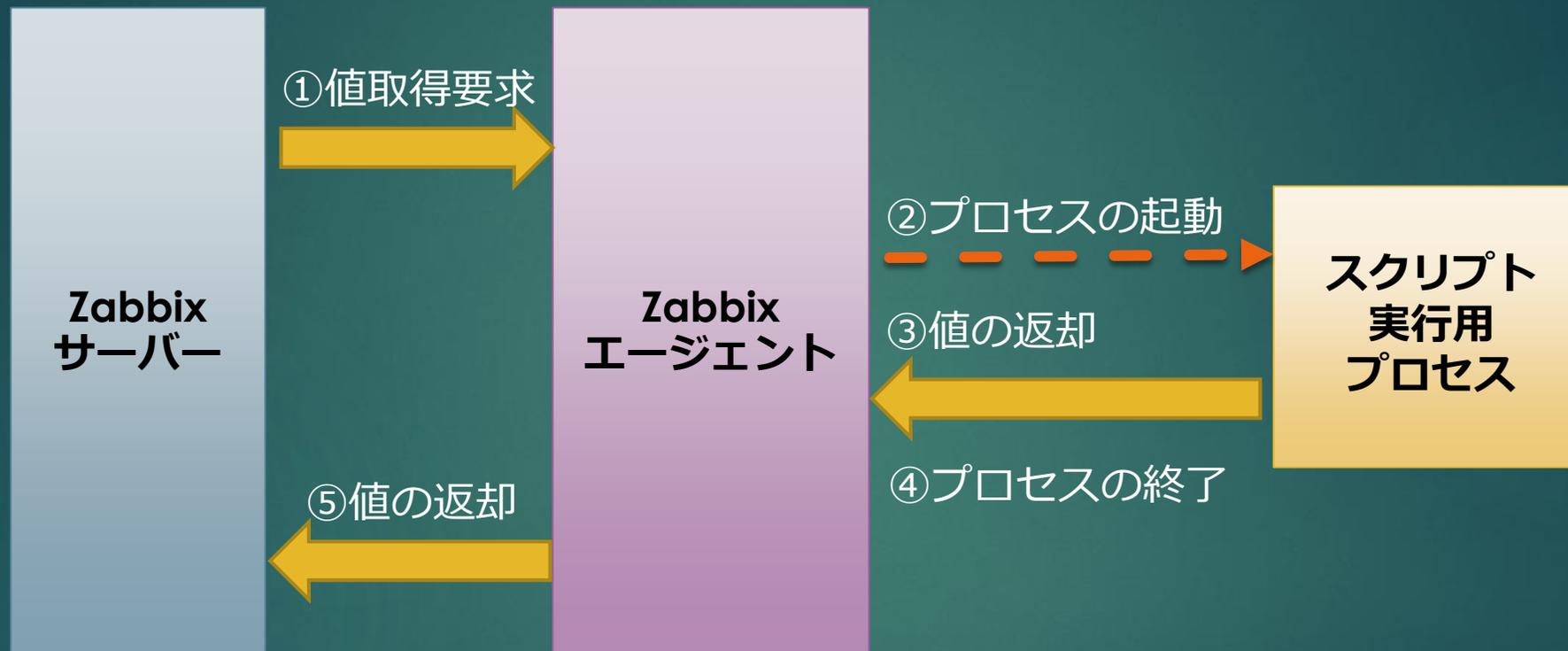
- ☹️ コンテナのステータス情報の値を取得するたびに、Docker関連のコマンドを実行したり、cgroup関連の仮想ファイルから値を切り出すコマンドやスクリプトを実行するんでしょ？
- ☹️ プロセスの起動終了って数が増えるとそれなりの負荷になるんだよ。わかってる？
- ☹️ パフォーマンス悪そ～。
- ◎ そんなあなたにZabbixの「ローダブルモジュール」

ローダブルモジュールとは

- ▶ Zabbix 2.2から用意された新機能
- ▶ C言語等で生成されたライブラリを動的にリンクして呼び出す機能
- ▶ 多数の値を取得する際、UserParameterなどの外部コマンドやスクリプトを実行する方法では子プロセス起動と終了の処理がボトルネックに
- ▶ ローダブルモジュールを利用すれば、呼び出し時にプロセスを別途起動することが不要
- ▶ 値取得の負荷を軽減し、より高パフォーマンスな情報収集が可能に！

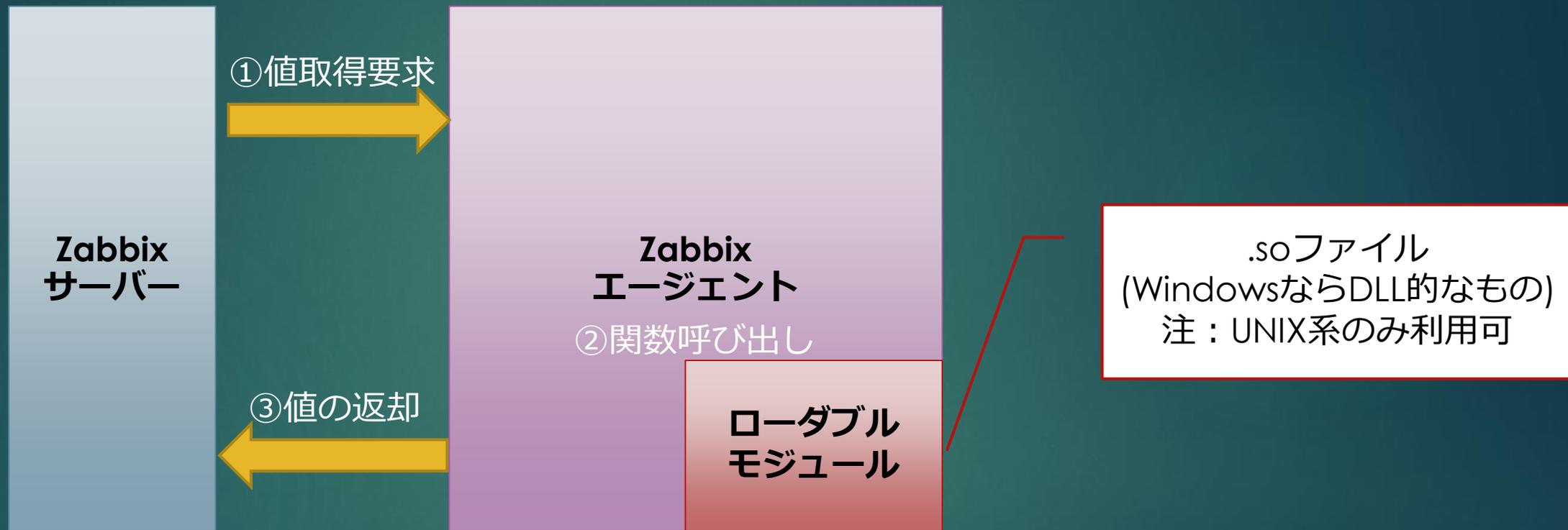
スクリプト呼び出し時の動作

21



プロセスを起動終了させる処理が多くなってくると、CPUへの負荷が増大する。

ロードダブルモジュールでの動作



ライブラリの関数の呼び出しだけで値を取得できるので、CPUへの負荷が低く、応答速度も速い。

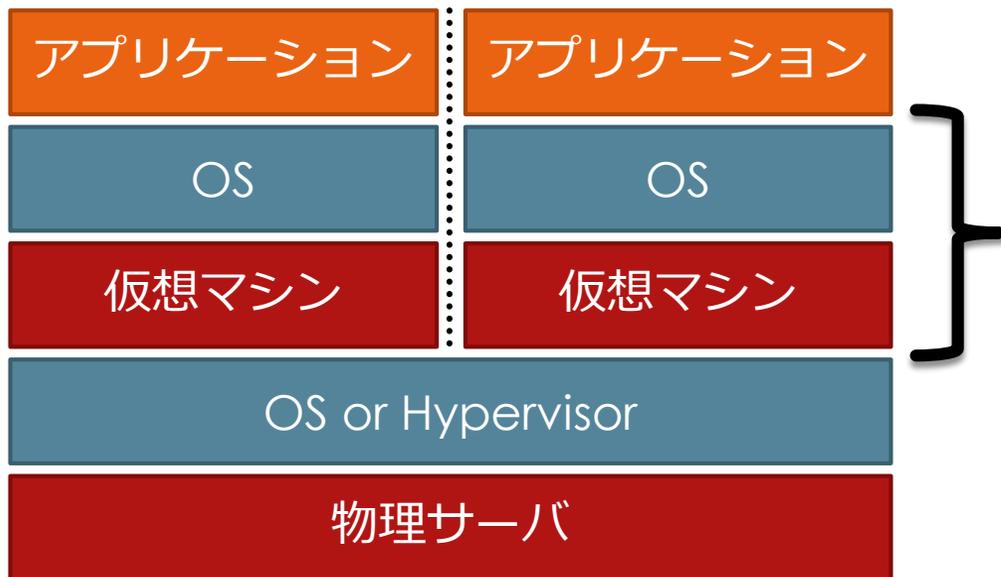
Dockerとは？

- ▶ Linux上では、以前からコンテナと呼ばれるOS上を論理的に分けて処理を実行される機能が用意されていた
- ▶ しかし、それを使いこなすことが難しかったため、あまり普及してこなかった
- ▶ Dockerは、コンテナ技術を簡単に利用できるように、コンテナとして機能を簡単に実装させたり、コンテナ同士や他のサーバとのネットワーク接続を設定することができる

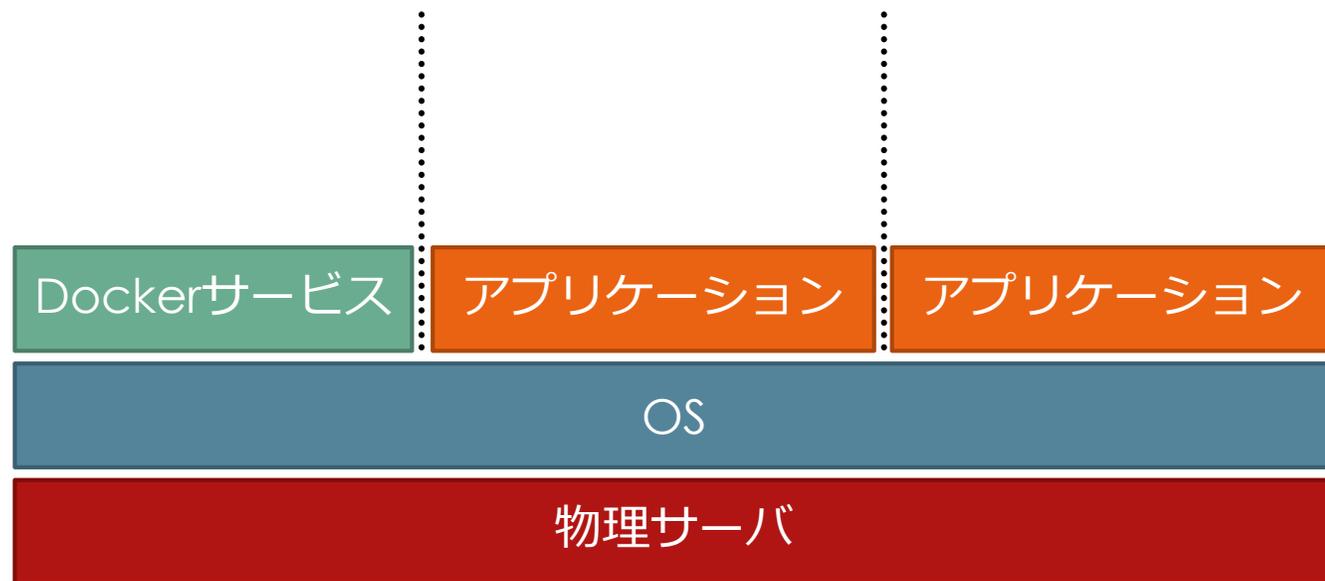
Dockerの構成イメージ

25

仮想化環境



Docker環境



Dockerの特徴

- ▶ Dockerfileというファイルに、コンテナの内容と構築する手順を記述しておくことで、同じ内容のコンテナを簡単に用意できる。
- ▶ コンテナのイメージをローカルのハードディスクにキャッシュできるので、同じイメージを利用するのであれば、同じ内容のコンテナを新規であっても数秒で起動できる。
- ▶ コンテナ上で稼働するアプリケーションであれば、可搬性が高くなり、開発環境と本番環境での環境に依存する影響を軽減できる。

ZabbixでDocker環境を監視

構想はあったものの...

- ▶ ローレベルディスクバリアやローダブルモジュールの機能を組み合わせて監視用のテンプレートを作成する構想はありました。
- ▶ 調査もしていて、どうやって各コンテナのリソース状態を取得するかもおおよそ特定できていました。
 - ▶ 例 : Runtime Metrics
<http://docs.docker.com/articles/runmetrics/>
- ▶ 作らなきゃな...

そうこうしているうちに

▶ 作ってくれた人が公開してくれました。

◎ 「Zabbix Docker Monitoring」

<https://github.com/monitoringartist/Zabbix-Docker-Monitoring>

これで何ができる

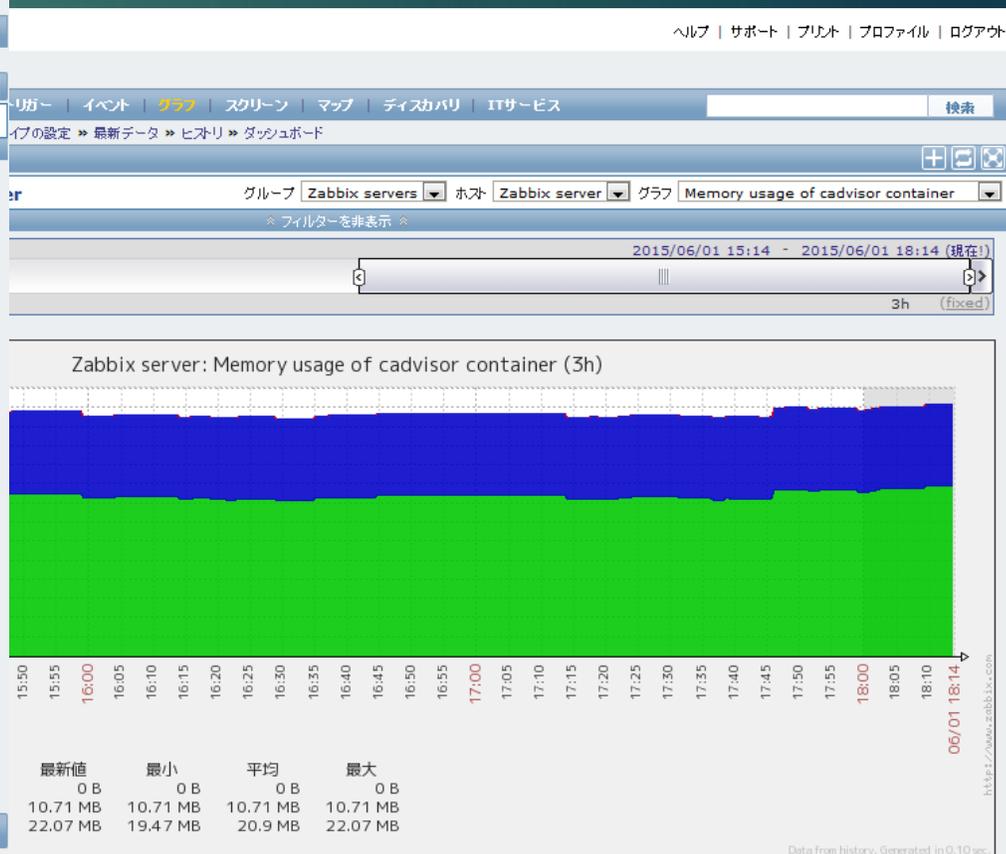
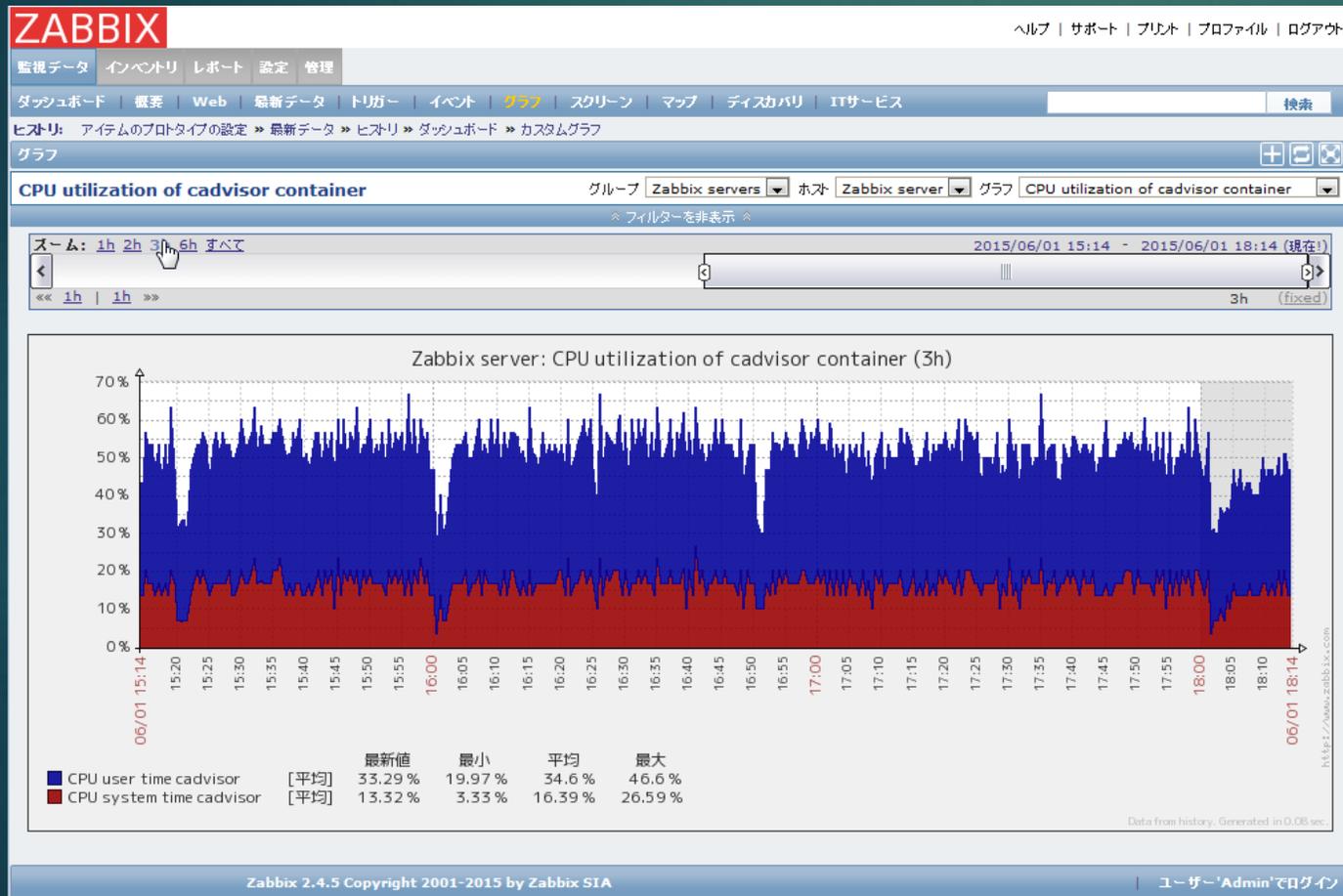
- ▶ これを利用すれば、以下のようなことがさくっと実現可
 - ▶ 作成されたコンテナのリストを自動取得
 - ▶ ローレベルディスクカバリを利用して、各コンテナのリソース情報を収集するアイテムが自動生成
 - ▶ ローダブルモジュールとして作られているので、それぞれの値取得はC言語のライブラリから直接取得(高速！)

こんな感じ

- ▶ 取得できるリソース情報としては以下のようなものが取得できます。
 - ▶ コンテナの起動状態
 - ▶ メモリ利用状況、スワップ発生量
 - ▶ CPU利用状況
 - ▶ ブロックデバイスに対するI/O
 - ▶ ネットワークトラフィック(実験的実装)

グラフも

▶ Zabbixなので、取得した情報はもちろんグラフに。



インストール手順 (CentOS 7上)

- ① Docker稼働しているサーバーにZabbixエージェントをインストールして、その後以下のコマンドを実行する。

```
# groupadd docker  
# chown root:docker /var/run/docker.sock  
# usermod -a -G docker zabbix
```

- ② 先のリポジトリから利用している環境に合ったライブラリを取得、もしくはソースをダウンロードして、ライブラリを自分でビルドし、モジュール用ディレクトリ (/usr/lib/zabbix/modules) にコピーする。
- ③ 通常のZabbixエージェントの設定(Hostname、Server、ServerActiveなど)を行う。
- ④ さらに、zabbix_agentd.confに以下の設定を追加する。

```
LoadModulePath=/usr/lib/zabbix/modules  
LoadModule=zabbix_module_docker.so
```

- ⑤ 設定が終わったら、Zabbixエージェントを起動または再起動
- ⑥ Zabbixサーバ側には、先のリポジトリからダウンロードしたテンプレートをインポートして、監視対象のホストに紐づければ監視開始。

まとめ

- ▶ Zabbixなら

- ▶ 「ローレベルディスクカバリ」で監視対象の増減に自動対応
- ▶ 「ローダブルモジュール」を活用して監視用の値取得の負荷を下げ、パフォーマンスを改善

いやいやZabbixは...

☹️ 見づらい！

☹️ 使いづらい！

☹️ UIが古い！

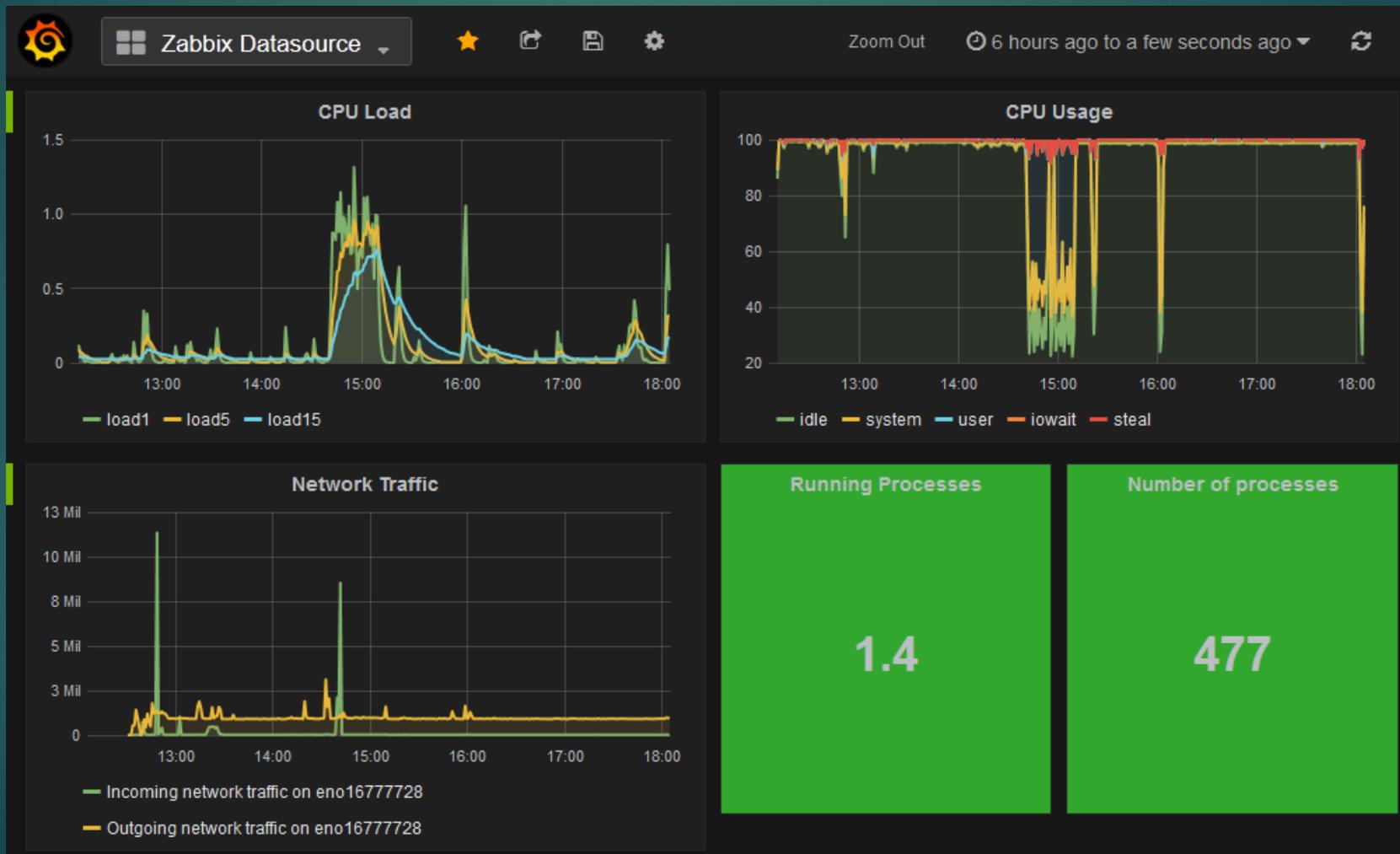
◎ そんなあなたに「Zabbix datasource for Grafana dashboard」！

<https://github.com/alexanderzobnin/grafana-zabbix>

こんな感じ

36

- ▶ こんな感じで見ることができます。



Zabbix 3.0について

Zabbix 3.0最新情報

38

- ▶ 先日(2015/09/11-12)のZabbix Conference 2015 @ラトビア でも発表がありました
が、現在、鋭意開発中です。
- ▶ 現時点では、Zabbix 3.0.0alpha3がリリースされています。
- ▶ 主な実装済機能 (2015/10/24現在)
 - ▶ SMTP authentication (これまでは認証なし25番ポート固定)
 - ▶ housekeeperの手動実行 (自分の好きなタイミングで実行可)
 - ▶ 予測値を取得する関数の追加 (線形、多項式、指数、対数、累乗近似などを利用)
 - ▶ アイテムの取得間隔のカスタマイズ (○時00分ちょうどなど指定可)
 - ▶ 各コンポーネント間の通信の暗号化と認証 (PSK、証明書などTLS 1.2に対応)
- ▶ 主な実装予定機能
 - ▶ Webインターフェースのデザインと内部実装の改善

Zabbix 3.0の画面イメージ

監視データ 監視データ インベントリ レポート 設定 管理

一般設定 プロ ホストグループ テンプレート **ホスト** メンテナンス アクション スクリーン スライドショー マップ ディスカバリ ITサービス

メディアタイプ ホスト

グループ

名前 DNS名 IPアドレス ポート

<input type="checkbox"/>	名前 ▲	アプリケーション	アイテム	トリガー	グラフ	ディスカバリ	タイプ	IPアドレス	ポート	Template	ステータス	ディスカバリ	エージェントの状態	情報	エージェント略号化
<input type="checkbox"/>	Zabbix server	アプリケーション 11	アイテム 75	トリガー 46	グラフ 13	ディスカバリ 2	Web	127.0.0.1:	10050	Template App Zabbix Server, Template OS Linux (Template App Zabbix Agent)	有効	ZBX SNMP JMX IPMI	PSK PSK		
<input type="checkbox"/>	zbx30	アプリケーション 10	アイテム 44	トリガー 19	グラフ 8	ディスカバリ 2	Web	10.34.16.128:	10050	Template OS Linux (Template App Zabbix Agent)	有効	ZBX SNMP JMX IPMI	PSK PSK		

0 選択

Zabbix 3.0利用時の注意点

40

- ▶ PHPのバージョンは5.4以降が必須
 - ▶ libcurlのバージョンが古いとSMTP authentication利用不可(7.20.0以上で利用可)
 - ▶ IE 8はサポート対象から削除
- ※ これは、アルファ版の時点(2015/10/24)での情報ですので、最終的なリリースバージョンでは変更される可能性があります。

▶ 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

- ☹ Zabbixがさあ、CentOS 7.1.1503で起動できないんだけど。
- ◎ CentOS 7.1のベースとなっているRHEL 7.1に不具合がありました。
- ◎ 7.1.1503リリース後に修正されたtrousersのパッケージ (trousers-0.3.11.2-4) がリリースされてますので、必ず最新のバージョンに更新してください。