



Zabbix as a framework

SoftLayer のクラウド環境を、 Zabbix で効率的に監視するには

Masahito Zembutsu @zembutsu

Technology Evangelist; Creationline, Inc.

Zabbix Conference Japan 2014, Tokyo, 21 Nov 2014

自己紹介

Why am I here?

ぜんぶつ まさひと
@zembutsu 前佛 雅人

➡ Technology Evangelist

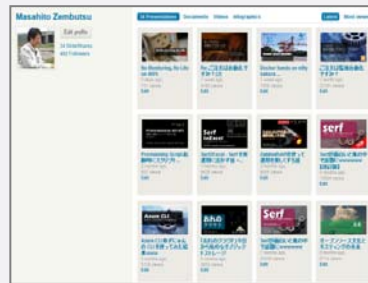
➡ Creationline, Inc

➡ 経歴

- 2000年4月 ~ ホスティング・ISP 事業者のサポート、業務全般
- 2013年12月 ~ Creationline で SoftLayer の構築支援/運用



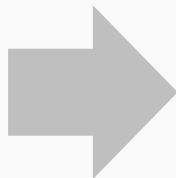
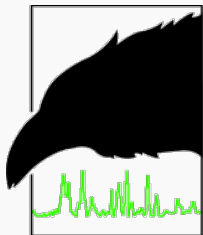
- オープンソース・コミュニティ参加
Eucalyptus User Group, OpenCloudCampus, ETC
- 趣味としての技術翻訳
(ICQ, Xen, Serf, Docker)
- スライド公開 (今日現在 34 本)
<http://slideshare.net/zembutsu/>



- 良いものを伝えたい人。

SoftLayerのクラウド環境を Zabbixで効率的に監視するには

Zabbix as framework and Effective Monitoring



SoftLayerのクラウド環境を
Zabbixで効率的に監視するには

Zabbix as framework and Effective Monitoring

本セッションは、クラウド時代に対応した運用と監視について、
Zabbix で効率的に行う方法をご紹介します。

クラウド環境の普及で、サーバが瞬時に手に入るようになりました。

しかし、環境構築や監視には時間がかかりがち。

しかも、クラウドでは環境が変化します。

クラウドの利点を活かしつつ、監視も迅速に行う方法を、
SoftLayer クラウドを例にして検討します。

<http://www.creationline.com/msp/>

MSP
Management Service Provider


CREATIONLINE, INC.
PLANNING. PROPOSAL. PRACTICE. FOR SUCCESS

わずらわしいサーバ管理から解放
24時間365日安心の監視・障害対応サービス

MSP
Management Service Provider

② MSPとは

■ MSPサービス・価格

✉ お問い合わせ

クリエーションラインでは、24時間365日サーバの運用・監視
約2000台以上の運用実績で培った信頼のサービスを提供いたします。

ステマ

SoftLayerのクラウド環境を
Zabbixで効率的に監視するには

Zabbix as framework and Effective Monitoring

MSP業務にZabbixを**何故導入**したのか？

どのように**拡張**していくのか？ 鍵は **API**

SoftLayerのクラウド環境を
Zabbixで効率的に監視するには

Zabbix as framework and Effective Monitoring

Zabbix as a **framework**

ソフトウェアフレームワーク(英: *software framework*)とは、プログラミングにおいて、一般的な機能をもつ共通コードをユーザーが選択的に上書きしたり特化させたりすることで、ある特定の機能をもたせようとする抽象概念のことである。単にフレームワークとも呼ばれる。

1. クラウドの価値とは？

➡ おさらい

2. 効率的な監視の検討

➡ なぜ Zabbix を導入したのか？

3. API は効率化の鍵

➡ Zabbix の API 入門

4. SoftLayer

➡ SoftLayer環境の監視をこうやっています

5. 実践SoftLayer

➡ ケース；クラウド破産を防げ！

SoftLayerのクラウド環境を
Zabbixで効率的に監視するには

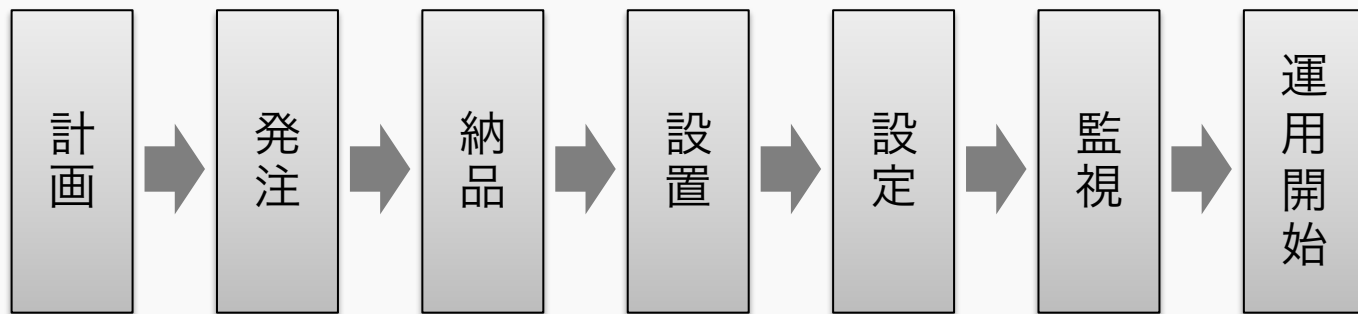
Zabbix as framework and Effective Monitoring

1 クラウドの価値とは？

The Value of Cloud Computing

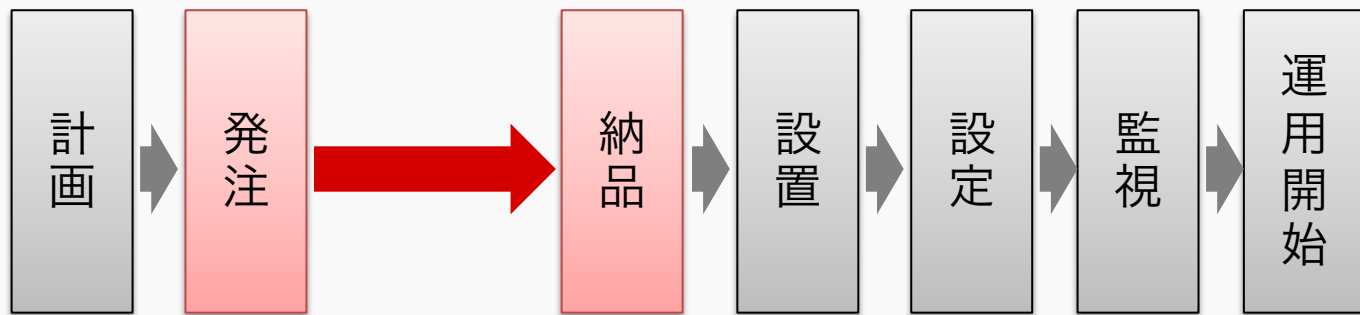


かつてのサーバ追加



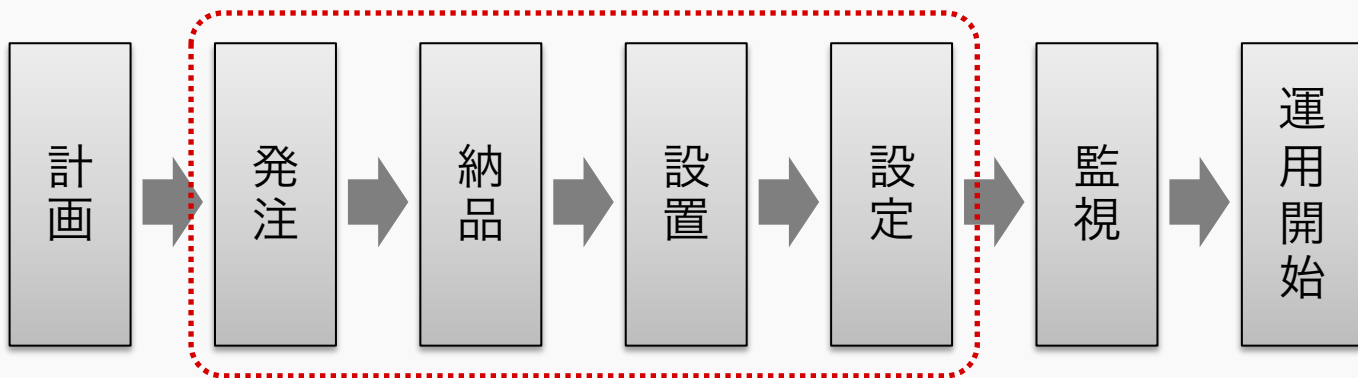
かつてのサーバ追加

実際のフローでは、ここがネックになりがち・・・



数週～1ヶ月程度
few weeks or a month

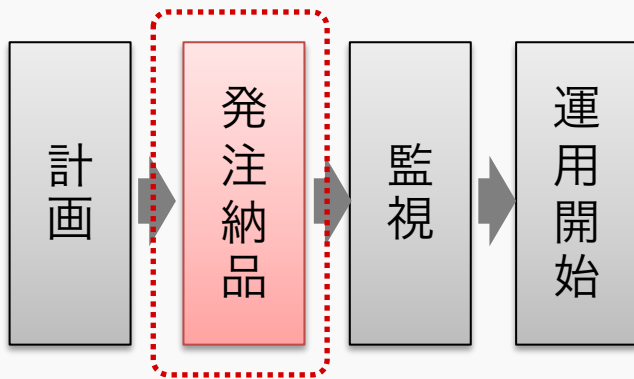
クラウド・コンピューティング基盤 Infrastructure as a Service



1 クリック
one click

クラウドでの追加

クラウド・コンピューティング基盤 **Infrastructure as a Service**



数分～10分
few minutes

クラウドの利点とは？

■ 技術的

- ➡ サーバの追加や削減が簡単に、今すぐできる
- ➡ 自動化・省力化のためのツールが提供されている

- ☞ 「柔軟性」「スケーラビリティ」
 - » “Flexible”
 - » “Agile”
- ☞ 使いやすいWebインターフェース
- ☞ API やライブラリ群

■ ビジネス的

- ➡ 利用時間単位の課金なので、
初期コストを抑えることができる

- ☞ 水道や電気料金のように、
必要な分を使った分だけ課金となる
 - » “On demand”
 - » “Pay as you go”
 - » “Pay for what you use”

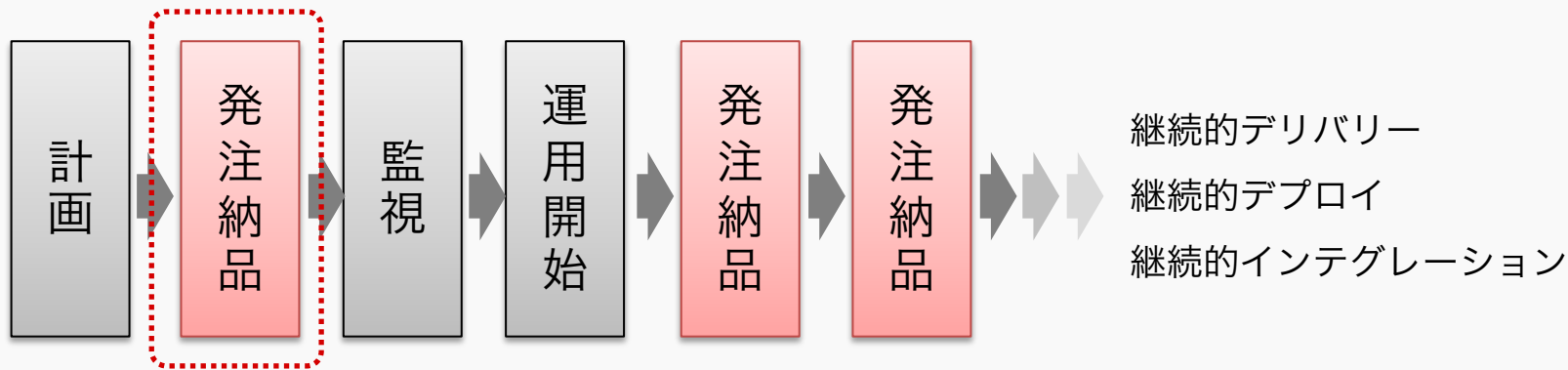
■ 背景

- ➡ ビジネスを取り巻く環境が常に変化する
- ➡ 周辺技術の進化（スマートフォンの急速な普及）

**最小コストでの最短スタートができ、
サービスに応じてインフラを変化できる**

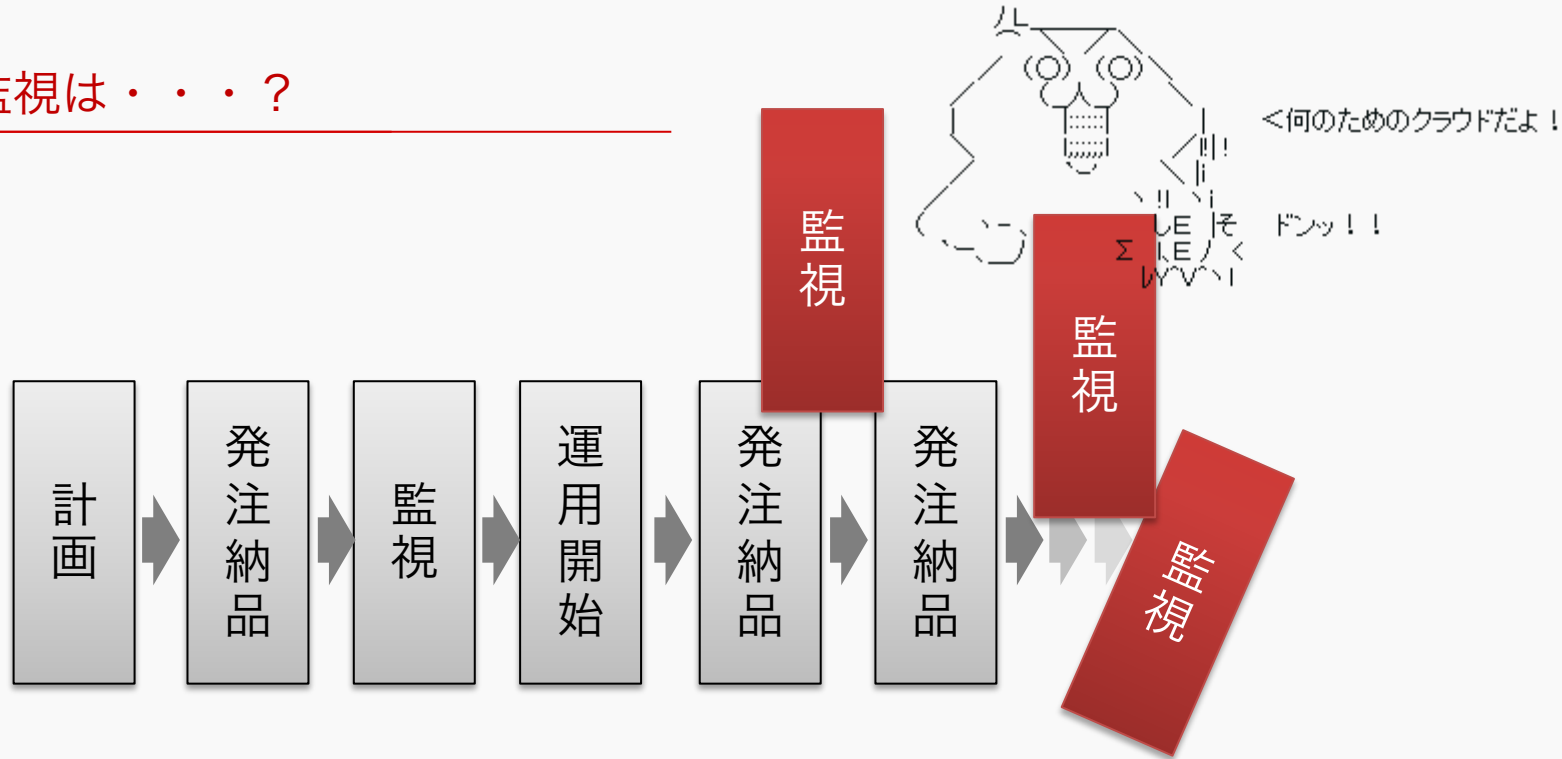
ところが・・・

クラウド・コンピューティング基盤 Infrastructure as a Service



数分～10分
few minutes

あれ、監視は・・・？



クラウドによって環境の準備が迅速になったはず。
しかし、台数の増加やシステム構成の複雑化によって、
クラウドの価値を相殺しているのではないのか？

SoftLayerのクラウド環境を
Zabbixで効率的に監視するには

Zabbix as framework and Effective Monitoring

2 効率的な監視の検討

What is effective operation and monitoring?

■ 運用の目的とは？

- ➡ サービスやシステム安定稼働（あるべき状態を維持）
- ➡ 異常があれば、復旧する
- ➡ そのために監視を行う

■ 監視とは？

- ➡ データを収集する
- ➡ データの異常を発見する
- ➡ 状況に応じて、適切な行動を起こす）

クラウド環境で求められる監視とは？

■ 監視設定の省力化

➡ 自動登録や API の仕組み

■ 短時間での状況把握と判断

➡ アラート

➡ グラフの一覧性

■ クラウド環境との連携

➡ 監視システムがAPIを持ち、クラウドのAPIも扱える

必要なものは、全てZabbixで揃っている

■ 運用前

➡ 監視設定の省力化？

- 役割に応じた監視の自動設定
- 追加・削除の自動化

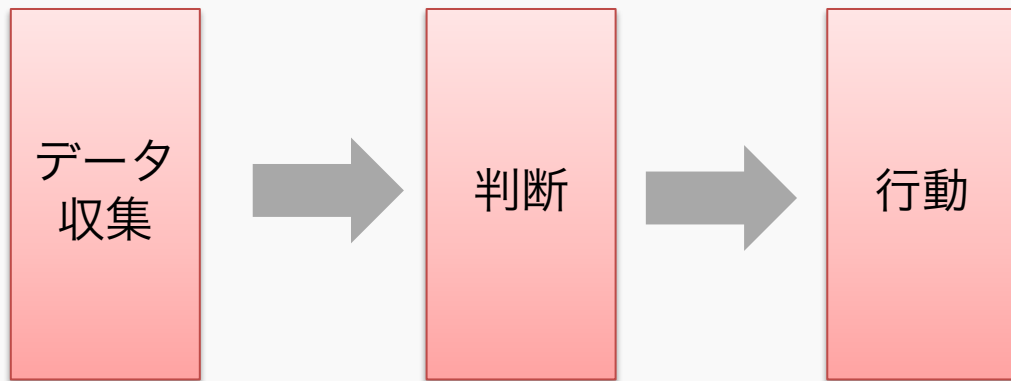
■ 運用後

➡ 迅速な検出とエスカレーション

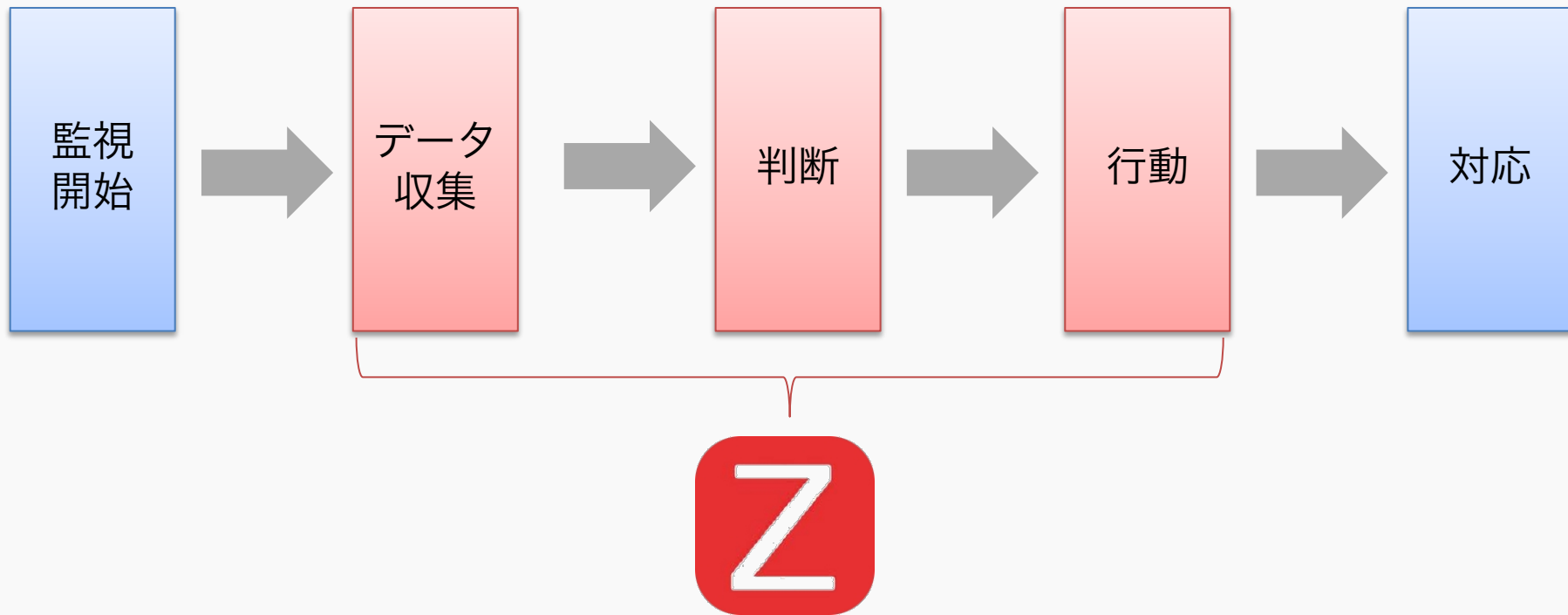
➡ 適切に「人間が」状況判断できる仕組みの提供

- トリアージ的な考え方の導入

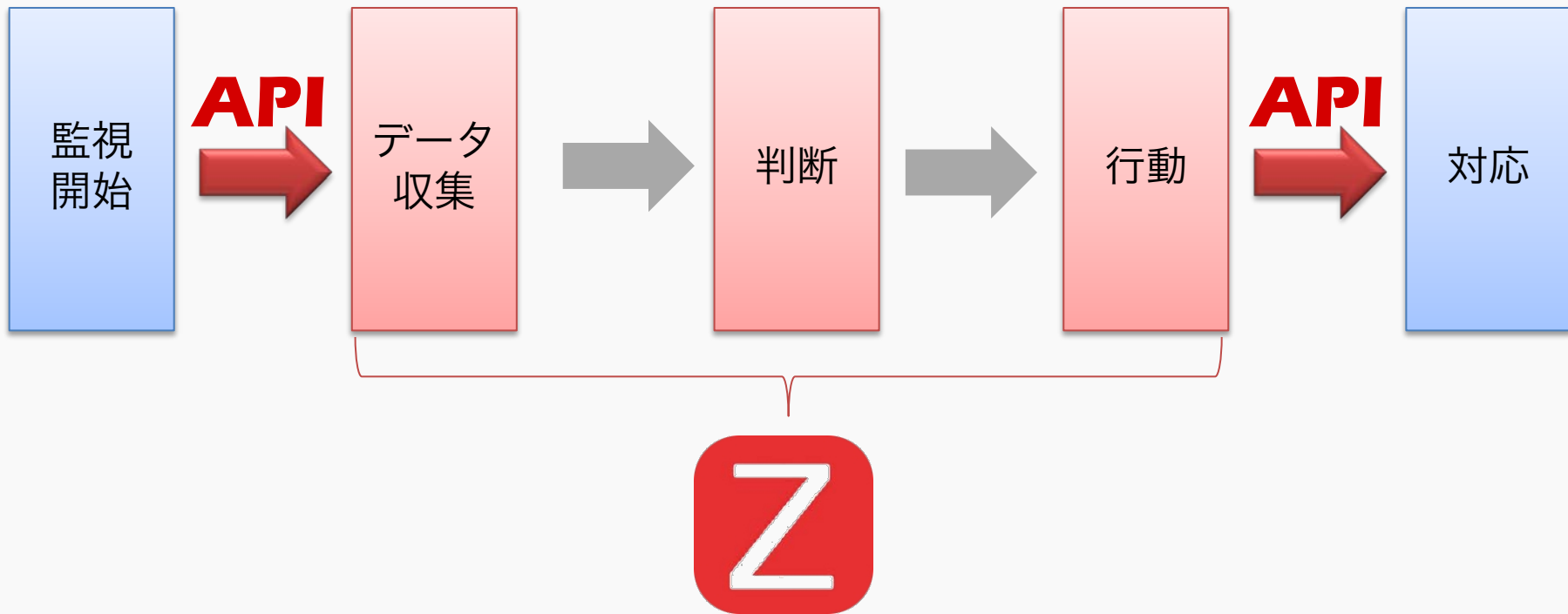
Zabbixを運用フレームワークと考える



Zabbixを運用フレームワークと考える



Zabbixを運用フレームワークと考える

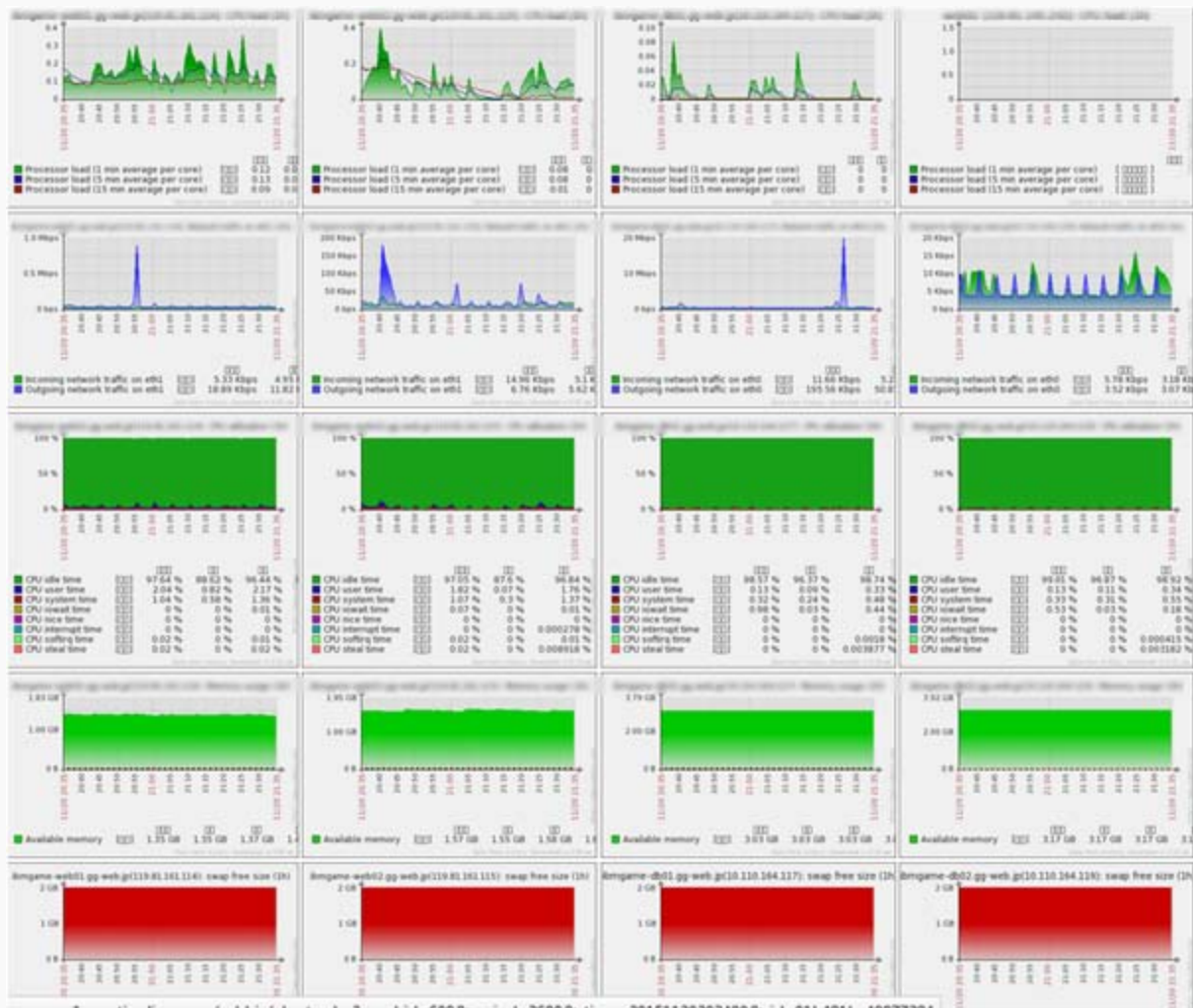


SoftLayerのクラウド環境を
Zabbixで効率的に監視するには

Zabbix as framework and Effective Monitoring

3 *APIは効率化の鍵*

An automation for practical operation



ZABBIX Help | Get support | Print | Profile | Logout

Monitoring | Inventory | Reports | **Configuration** | Administration

Host groups | Templates | Hosts | Maintenance | Actions | **Screens** | Slide shows | Maps | Discovery | IT services

History: スクリーンの設定 » ユーザープロフィール » Configuration of screens » User profile » Configuration of screens

CONFIGURATION OF SCREENS

serer status

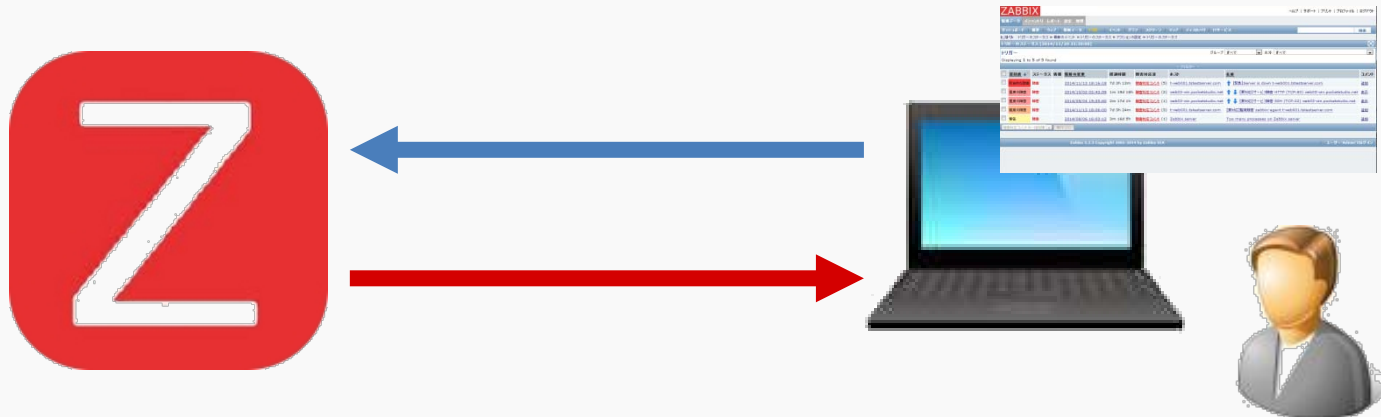
<p>10.110.42.194: Memory usage (1h)</p> <p>11/20 10:50 PM 10:55 PM 11:00 PM 11:05 PM 11:10 PM 11:15 PM 11:20 PM 11:25 PM 11:30 PM 11:35 PM 11:40 PM 11:45 PM 11/20 11:50 PM</p> <p>■ Total memory [no data] ■ Available memory [no data]</p> <p>last min avg max</p> <p>Data from history. Generated in 0.03 sec.</p> <p>Change</p>	<p>Screen cell configuration</p> <p>Resource <input type="text" value="Graph"/></p> <p>Graph name <input type="text" value="munin.jp: CPU utilization"/> Select</p> <p>Width <input type="text" value="500"/></p> <p>Height <input type="text" value="100"/></p> <p>Horizontal align <input type="button" value="Left"/> <input checked="" type="button" value="Center"/> <input type="button" value="Right"/></p> <p>Vertical align <input type="button" value="Top"/> <input checked="" type="button" value="Middle"/> <input type="button" value="Bottom"/></p> <p>Column span <input type="text" value="1"/></p> <p>Row span <input type="text" value="1"/></p> <p>Dynamic item <input type="checkbox"/></p> <p>Save Cancel</p>	<p>Change</p>
<p>Change</p>	<p>Change</p>	<p>Change Change Change</p>
<p>Change</p>	<p>Change</p>	<p>Change Change Change</p>

1. 【設定】 → 【スクリーン】
→ 【対象スクリーン名】 をクリック
2. 対象枠の【変更】 をクリック
3. グラフ名の【選択】 ボタンをクリック
4. グループとホストを選び、アイテム選択
5. 必要に応じて列や高さの調整
6. 保存
7. 必要アイテム・ホスト分2～6を繰り返す



お前それクラウドでも
同じ事言えんの？

ZabbixのAPIを使って自動化するシステムのイメージ

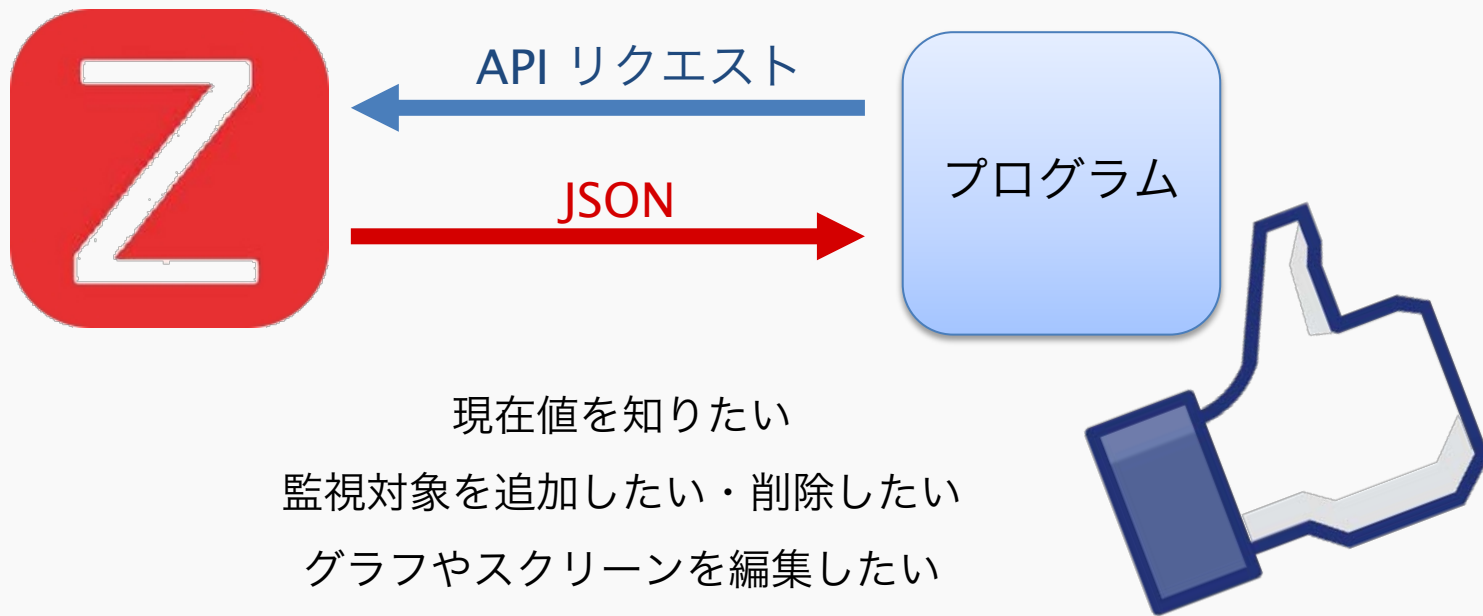


現在値を知りたい

監視対象を追加したい・削除したい

グラフやスクリーンを編集したい

ZabbixのAPIを使って自動化するシステムのイメージ



重要な点は、設定時間短縮・人的作業事故を回避

**今日から始めるZabbix API
APIで設定を追加するには？**

ZabbixのAPI ?

■ API は標準で使える機能

- ➡ HTTP を使って ZABBIX サーバに処理を要求
- ➡ JSON 形式でデータをやりとり
 - JSON-RPC



参考 : Getting started with Zabbix API

<http://blog.zabbix.com/getting-started-with-zabbix-api/1381/>

■ API = Application Programming Interface

- ➡ ソフトウェアが相互にデータをやりとりする
インターフェースの仕様
 - 関数
 - プロシージャール
 - 変数
 - データ構造

■ RPC = Remote Procedure Call



起源

RFC 707 (1976年発表)

- ➡ “プログラムから別のアドレス空間（通常、共有ネットワーク上の別のコンピュータ上）にあるサブルーチンや手続きを実行する事を可能にする技術。遠隔鉄続き呼び出しとも。その際に、遠隔相互作用の詳細を明示的にコーディングする必要が無い。つまり、プログラマはローカルなサブルーチン呼び出しと基本的に同じコードをリモート呼び出しについても行う。”

■ プロトコル

- ➡ 仕様上の基本機能
 - 定義した手続きの呼び出しはユニークである
 - リクエストメッセージに対応するレスポンスメッセージ提供
 - サービスの呼び出しや、呼び出されに応答する認証
- ➡ ポイントは、
 - リモート上のプログラムに対し、実行する手続きと引数を要求（コール）する。
- ➡ 様々な仕様
 - JSON-RPC 2.0は Zabbix が採用



■ JSON-RPC 2.0

➡ <http://www.jsonrpc.org/specification>

- JSON-RPC is a stateless, light-weight remote procedure call (RPC) protocol.
(1つの通信において以前の状態を保持しない)

➡ リクエストオブジェクトの仕様

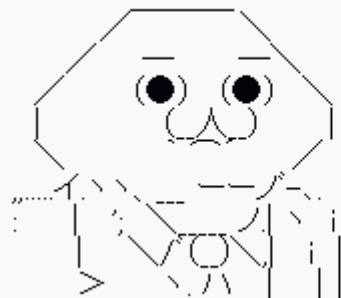
- 'jsonrpc' … JSON-RPC で“2.0”と明示の必要あり
- 'method' … 呼び出されるメソッド名
- 'param' … パラメータ (変数など)
- 'id' … 整数または NULL

➡ レスポンスオブジェクトの仕様

- 'jsonrpc' … JSON-RPC で“2.0”と明示の必要あり
- 'result' … 正常処理時は必須
- 'error' … 以上処理時は必須
- 'id' … 必須。ただしエラー時は NULL に



ZABBIXのデータの流れ



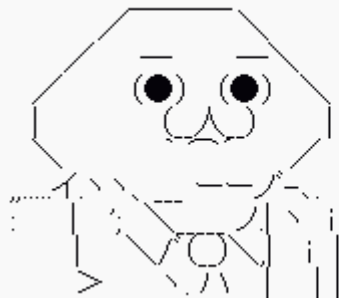
ID: admin 、 Pass: *** です

Zabbix Server



ZABBIXのデータの流れ

Zabbix Client



ID: admin、Pass: *** です

user.login

HTTP経由

Zabbix Server

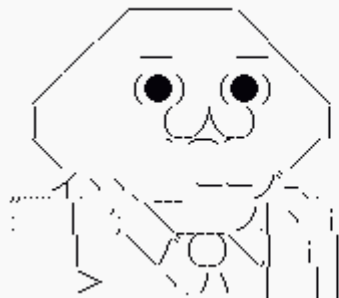


【 user.login】

<https://www.zabbix.com/documentation/2.2/manual/api/reference/user/login>

ZABBIXのデータの流れ

Zabbix Client



認証トークン

ID: admin、Pass: *** です

Auth ID (sessionid)

Zabbix Server

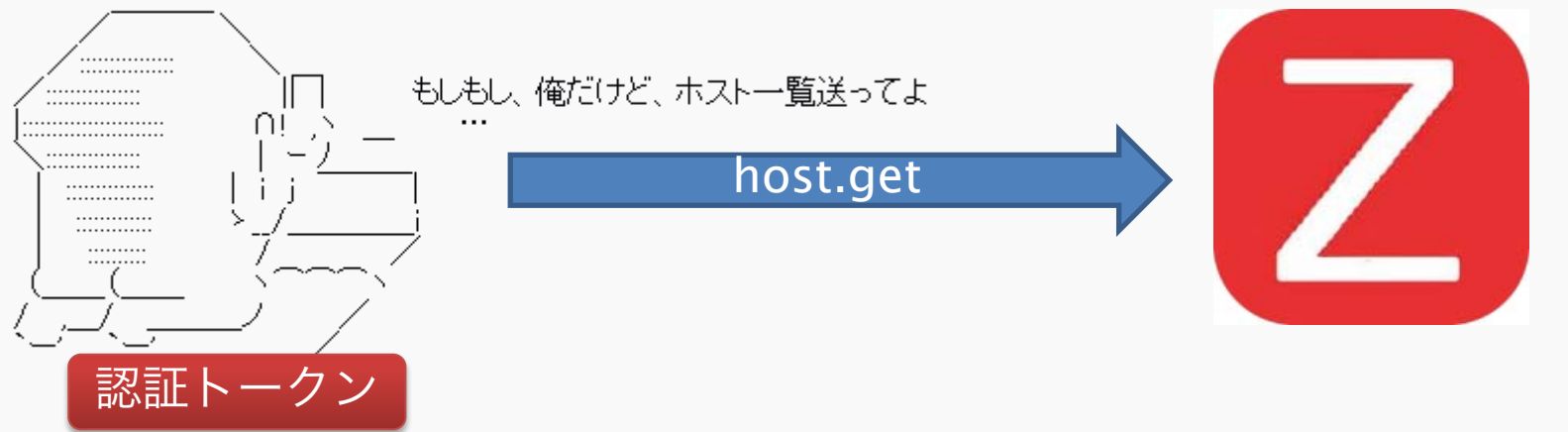


よし、通れ！

【 user.login】

<https://www.zabbix.com/documentation/2.2/manual/api/reference/user/login>

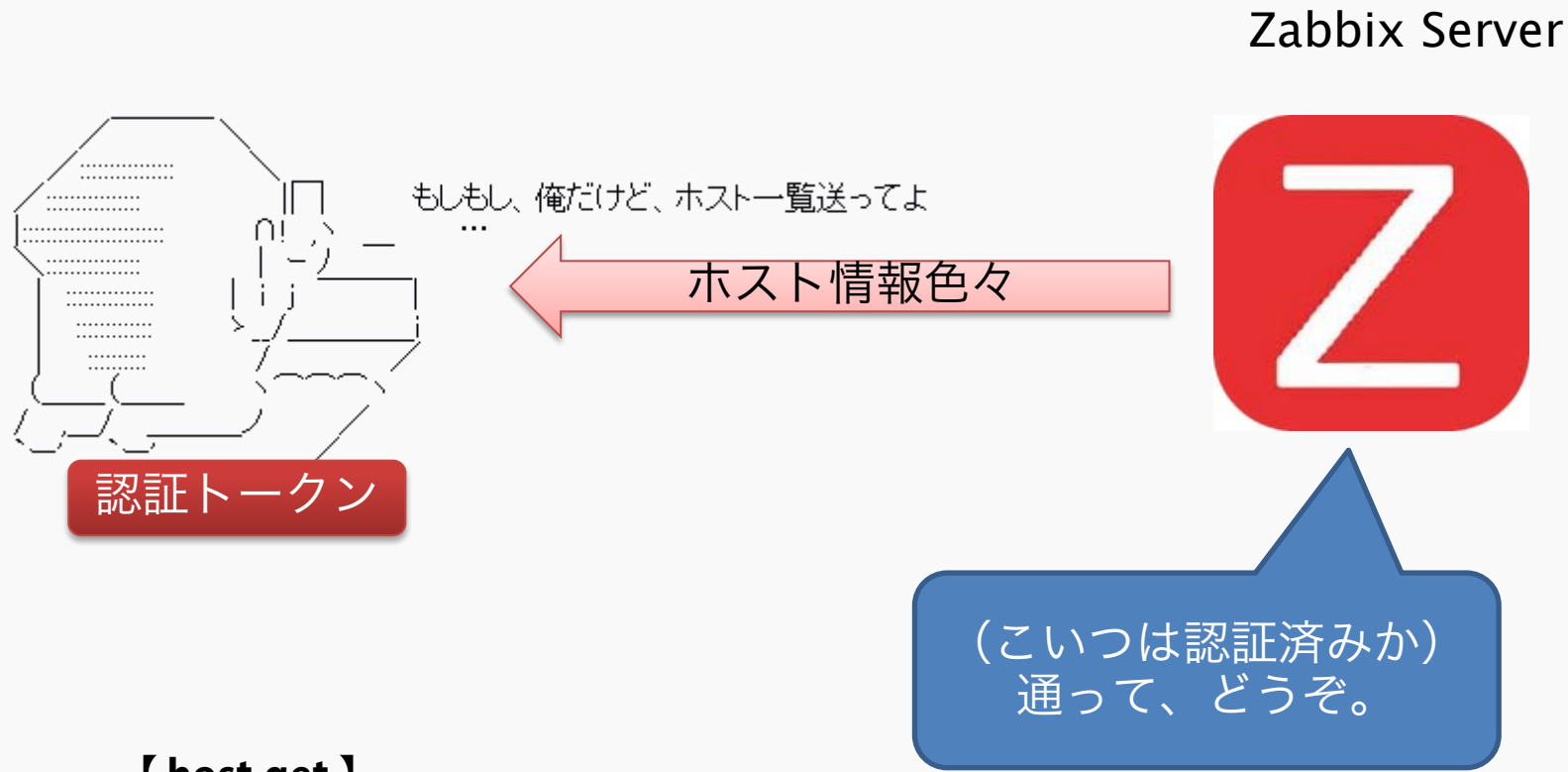
ZABBIXのデータの流れ



【 host.get 】

<https://www.zabbix.com/documentation/2.2/manual/api/reference/host/get>

ZABBIXのデータの流れ



【 host.get 】

<https://www.zabbix.com/documentation/2.2/manual/api/reference/host/get>

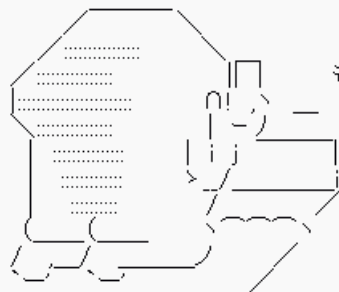
一連の流れをまとめると

Zabbix Client



ID: admin、Pass: *** です

認証トークン



もしもし、俺だけど、ホスト一覧送ってよ
...

Zabbix Server



user.login

Auth ID (sessionid)

host.get

ホスト情報色々

user.loginを手動で試す… 前に

■ そのまえに必要なもの

➡ Linux とかのコンソール

➡ cURL (curl コマンド)

- <http://curl.haxx.se/>
- コマンドラインで HTTP 通信出来る
yum install curl

➡ jq

- <http://stedolan.github.io/jq/>
- JSON の grep や sed/awk みたいなもの
yum install jq

👉 Mac OS の Terminal や Windows の Cygwin でも大丈夫のはず…



./jq

■ user.login とは

- ➡ API でログインし、認証用トークンを作成
- ➡ このトークンを使って、次に実際の処理を要求

■ パラメータ

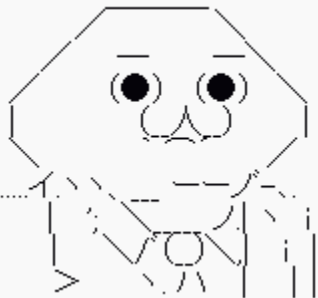
- ➡ user … Zabbix のユーザ名 (必須)
- ➡ password … パスワード (必須)
- ➡ userData … 詳細情報表示を表示するか
(true または false , オプション)

■ 戻り値

- ➡ result … 認証用トークン

```
{  
  "id": 1,  
  "params": {  
    "password": "zabbix",  
    "user": "admin"  
  },  
  "method": "user.login",  
  "jsonrpc": "2.0"  
}
```


user.login



ID: admin 、 Pass: *** です

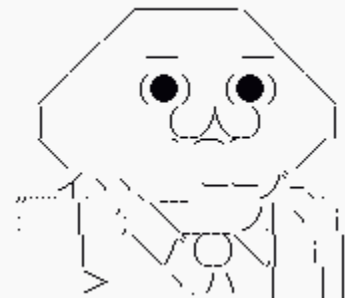
#!/bin/sh

```
{  
  "id": 1,  
  "params": {  
    "password": "zabbix",  
    "user": "admin"  
  },  
  "method": "user.login",  
  "jsonrpc": "2.0"  
}
```

json='{ "jsonrpc": "2.0", "method": "user.login", "params": { "user": "admin",
"password": "zabbix" }, "id": 1 }'

curl -s -XGET -H "Content-Type:application/json-rpc" ¥
-d \$json http://127.0.0.1/zabbix/api_jsonrpc.php | jq -r '.result'

user.login



ID: admin、Pass: *** です

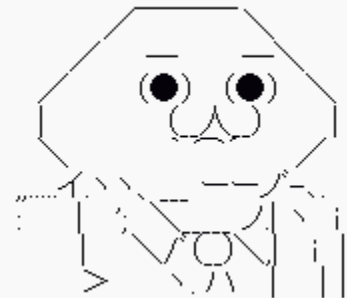
\$./auth.sh

```
{"id":1,"result":"4ffc67c712229cd683ae50fc05ae28c5","jsonrpc": "2.0"}
```

※スクリプトは Github に置きました。こちらをご利用ください。

<https://github.com/zembutsu/zabbix-api-tools/blob/master/auth.sh>

user.login



ID: admin 、 Pass: *** です

```
$ ./auth.sh | jq ‘.’
```

```
{
```

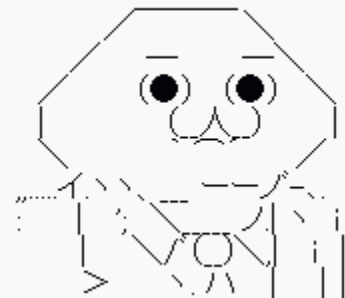
```
  "id": 1,
```

```
  "result": "4ffc67c712229cd683ae50fc05ae28c5",
```

```
  "jsonrpc": "2.0"
```

```
}
```

user.login



ID: admin 、 Pass: *** です

```
$ ./auth.sh | jq '.result'
```

```
4ffc67c712229cd683ae50fc05ae28c5
```

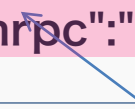
host.get

curl -s -XGET ¥

-H "Content-Type:application/json-rpc" ¥

-d '{"auth":"cc2590561f7474cfb613c934a45a69c4","id":2,"params":{"output":"extend"},"method":"host.get","jsonrpc":"2.0"}' ¥

http://127.0.0.1/zabbix/api_jsonrpc.php



```
{  
  "jsonrpc": "2.0",  
  "method": "host.get",  
  "params": {  
    "output": "extend"  
  },  
  "id": 2,  
  "auth": "###TOKEN###"  
}
```

host.get

```
curl -s -XGET ¥
```

```
-H "Content-Type:application/json-rpc" ¥
```

```
-d
```

```
{'auth':"cc2590561f7474cfb613c934a45a69c4","id":2,"params":{"output":"extend"},"method":"host.get","jsonrpc":  
"2.0"}' ¥
```

```
http://127.0.0.1/zabbix/api_jsonrpc.php
```

```
{"jsonrpc":"2.0","result":[{"maintenances":[],"hostid":"10084","proxy_hostid":"0","host":"Zabbix  
server","status":"0","disable_until":"0","error":"","available":"1","errors_from":"0","lastaccess":"0","ipmi_authtype":"-  
1","ipmi_privilege":"2","ipmi_username":"","ipmi_password":"","ipmi_disable_until":"0","ipmi_available":"0","snmp_disa  
ble_until":"0","snmp_available":"0","maintenanceid":"0","maintenance_status":"0","maintenance_type":"0","maintenan  
ce_from":"0","ipmi_errors_from":"0","snmp_errors_from":"0","ipmi_error":"","snmp_error":"","jmx_disable_until":"0","j  
mx_available":"0","jmx_errors_from":"0","jmx_error":"","name":"Zabbix  
server","flags":"0","templateid":"0"},{"maintenances":[],"hostid":"10195","proxy_hostid":"0","host":"Overview  
Layer","status":"0","disable_until":"0","error":"","available":"0","errors_from":"0","lastaccess":"0","ipmi_authtype":"-  
1","ipmi_privilege":"2","ipmi_username":"","ipmi_password":"","ipmi_disable_until":"0","ipmi_available":"0","snmp_disa  
ble_until":"0","snmp_available":"0","maintenanceid":"0","maintenance_status":"0","maintenance_type":"0","maintenan  
ce_from":"0","ipmi_errors_from":"0","snmp_errors_from":"0","ipmi_error"
```

host.get

```
curl -s -XGET ¥
```

```
-H "Content-Type:application/json-rpc" ¥
```

```
-d
```

```
{ "auth": "cc2590561f7474cfb613c934a45a69c4", "id": 2, "params": { "output": "extend" }, "method": "host.get", "jsonrpc": "2.0" } ¥
```

```
http://127.0.0.1/zabbix/api_jsonrpc.php | jq '.{
```

```
"id": 2,
```

```
"result": [
```

```
{
```

```
  "templateid": "0",
```

```
  "ipmi_available": "0",
```

```
  "ipmi_disable_until": "0",
```

```
  "ipmi_password": "",
```

```
  "ipmi_username": "",
```

```
  "ipmi_privilege": "2",
```

```
  "ipmi_authtype": "-1",
```

```
  "lastaccess": "0",
```

```
  "errors_from": "0",
```

```
  "maintenances": [],
```

まとめ

■ API をコマンドラインで試すには

➡ curl と jq を使いこなそう

👉 Perl や Ruby や PHP など、プログラミングの知識や経験がなくても試せます。

■ どんな API があるの？

➡ Zabbix にかかわること、ほぼ全て

- <https://www.zabbix.com/documentation/2.2/manual/api>

■ あとは、試してみましょう ♪ ♪

冴えるシェル芸！ 敷居はさほど高くない…ハズッ

おまけ： ./json2zabbix.sh

```
#!/bin/bash
```

```
json_auth='{"jsonrpc":"2.0","method":"user.login","params":{"user":"admin","password":"zabbix"},"id":1}'  
rpc_url='http://127.0.0.1/zabbix/api_jsonrpc.php'
```

```
ZABBIX_TOKEN=`curl -s -XGET -H "Content-Type:application/json-rpc" ¥  
-d $json_auth $rpc_url | jq -r '.result'`
```

```
#echo "##### TOKEN = $ZABBIX_TOKEN"
```

```
if [ $1 ]; then
```

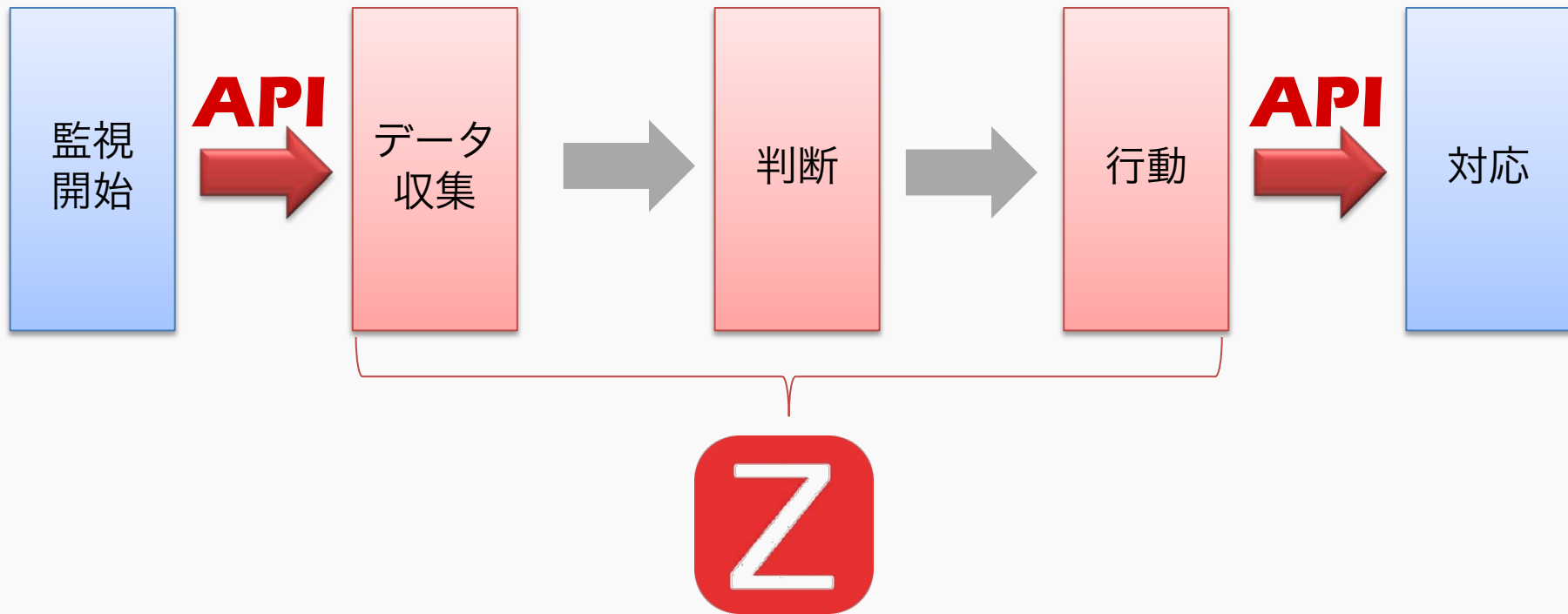
```
  json=`cat $1 | jq '.' -c | sed -e s/##TOKEN##/$ZABBIX_TOKEN/`  
  curl -s -XGET -H "Content-Type:application/json-rpc" ¥  
    -d $json $rpc_url
```

```
else
```

```
  echo "JSON to ZABBIX Request"  
  echo "Usage: json2zabbix.sh <filename>"
```

```
fi
```

Zabbixを運用フレームワークと考える



SoftLayerのクラウド環境を
Zabbixで効率的に監視するには

Zabbix as framework and Effective Monitoring

4 *SoftLayer*

SoftLayer's architecture and Network

SoftLayerとは？

■ 仮想化は選択肢の1つ

- ➡ 仮想サーバ、物理サーバー(BareMetal)を選択できる
- ➡ 時間単位の課金
- ➡ リソースは、共有と専有を選べる

■ 歴史

- ➡ 2005年 テキサス州ダラスに設立
- ➡ 2011年 アジア・欧米に展開開始
- ➡ 2013年 IBM社に買収
- ➡ 2014年 140ヶ国、25000以上の利用者
東京データセンター開設予定

SoftLayerが提供するサービス

■ サーバ

➡ 仮想サーバ、物理サーバ(baremetal)

■ ストレージ

➡ オブジェクトストレージ、NAS、iSCSI、CDN、バックアップ

■ ネットワーク

➡ VLAN、ロードバランサ、ネットワーク・アプライアンス、DNSサービス

■ セキュリティ

➡ ファイアウォール、SSL証明書、コンプライアンス

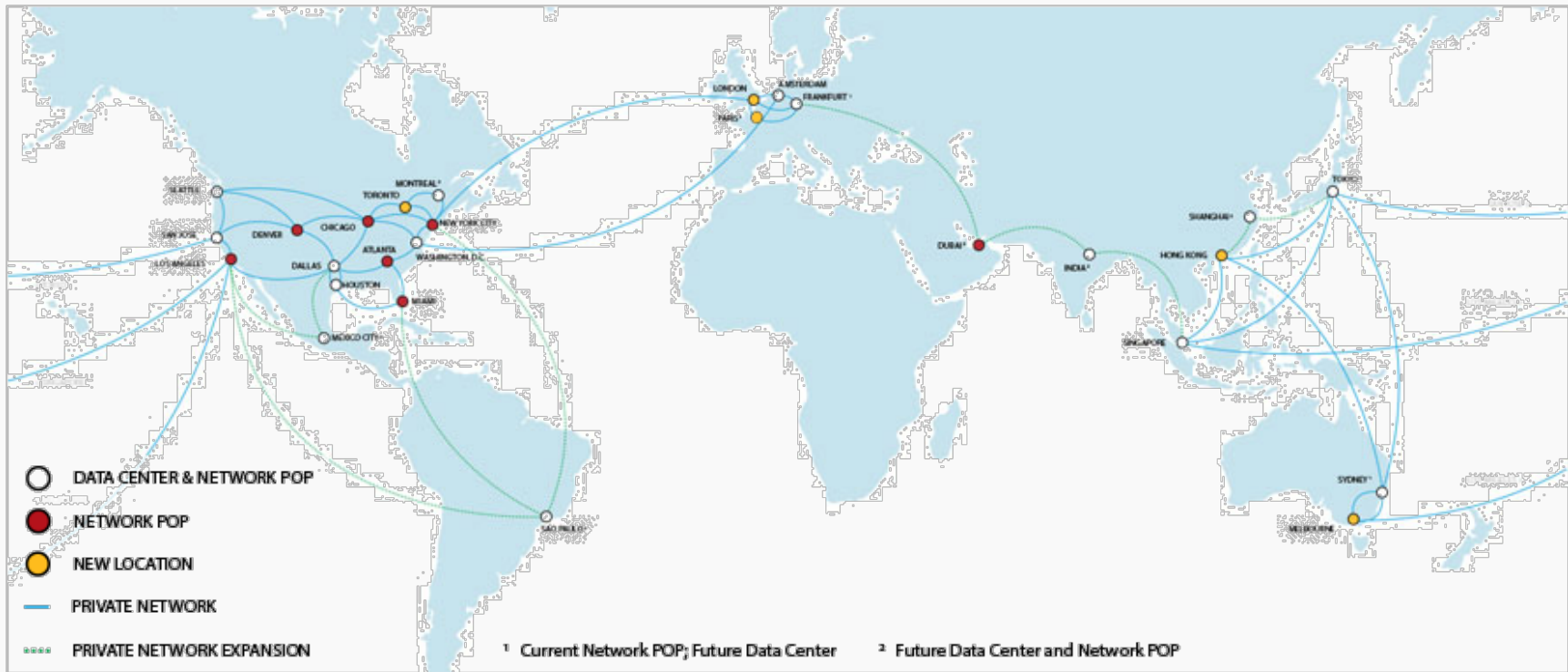
■ 開発者向け機能

➡ API、メール配信、メッセージキュー、フォーマット変換、サポート

SoftLayerのデータセンター

Global footprint.

A seamless platform that goes across any border.



SoftLayerのサポート

■ 料金

➡ 基本的なサポートは無料、制限なし

■ サポートの種類

➡ チケット

➡ チャット

- ・ リアルタイムで担当者とやりとり

➡ 電話

- ・ 現状は英語のみ

The screenshot displays the SoftLayer support portal interface. At the top, the user is logged in as 'Masahito Zembutsu' from 'Creationline Inc.' with ID '302402'. The navigation bar includes links for Support, Devices, Storage, Network, Security, Services, and Account. The main section is titled 'Tickets' and shows a list of tickets. The selected ticket, #14915524, is titled 'SNG01 - SCHEDULED Maintenance - Object Storage System - SNG01 - 11...'. The ticket details show a message from the System regarding network maintenance on November 18, 2014, at 10:34 pm. The message states that network engineers completed emergency maintenance to allow connectivity to 174.133.118.176 from hosts outside the SoftLayer network, and that the IP address for sng01.objectstorage.softlayer.net was changed during this maintenance. It also mentions that access to the new address is being blocked at the SoftLayer backbone and that network engineers are investigating.

Ticket #	Group	Subject/Title	Owner	Last Replied	Device	Updated
14915524	NetMaintenance	SNG01 - SCHEDULED Maintenance - Object Storage System - SNG01 - 11...	SL302402	Auto	None	47 minut...

System

Network Engineers completed an emergency maintenance to allow network connectivity to 174.133.118.176 from hosts outside of the SoftLayer network at 18-Nov-2014 12:53 UTC. This change has corrected the access issue to sng01.objectstorage.softlayer.net from the public internet.

If you are experiencing problems accessing sng01.objectstorage.softlayer.net please contact our support team via email, ticket or chat.

Thank you
Network Operations

[Update 3]

November 18, 2014, 10:34 pm

System

Some Object Storage customers are unable to access sng01.objectstorage.softlayer.net from host addresses outside of the SoftLayer network after the caching DNS servers, these hosts are using, refreshed their record for the name "sng01.objectstorage.softlayer.net". The IP address for sng01.objectstorage.softlayer.net was changed during this maintenance and it appears that access to the new address, of 174.133.118.176, is being blocked at the SoftLayer backbone. Network Engineers are investigating.

Thank you.
Network Operations

[Update 2]

November 18, 2014, 8:38 pm

■ 標準提供される2種類の監視

➡ BASIC Monitoring

- PING レベルの死活監視（無償）
- メール、チケットでの通知（無償）
- TCP Port レベルの監視オプションあり
- 障害発生時に自動再起動や、データセンタ有人対応

➡ Advance Monitoring

- 分単位のサービス・リソース監視
- Basic は無償
- AdvanceやPlatinumは有償

Zabbixの出番

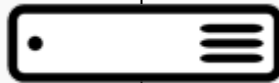


Virtual Servers

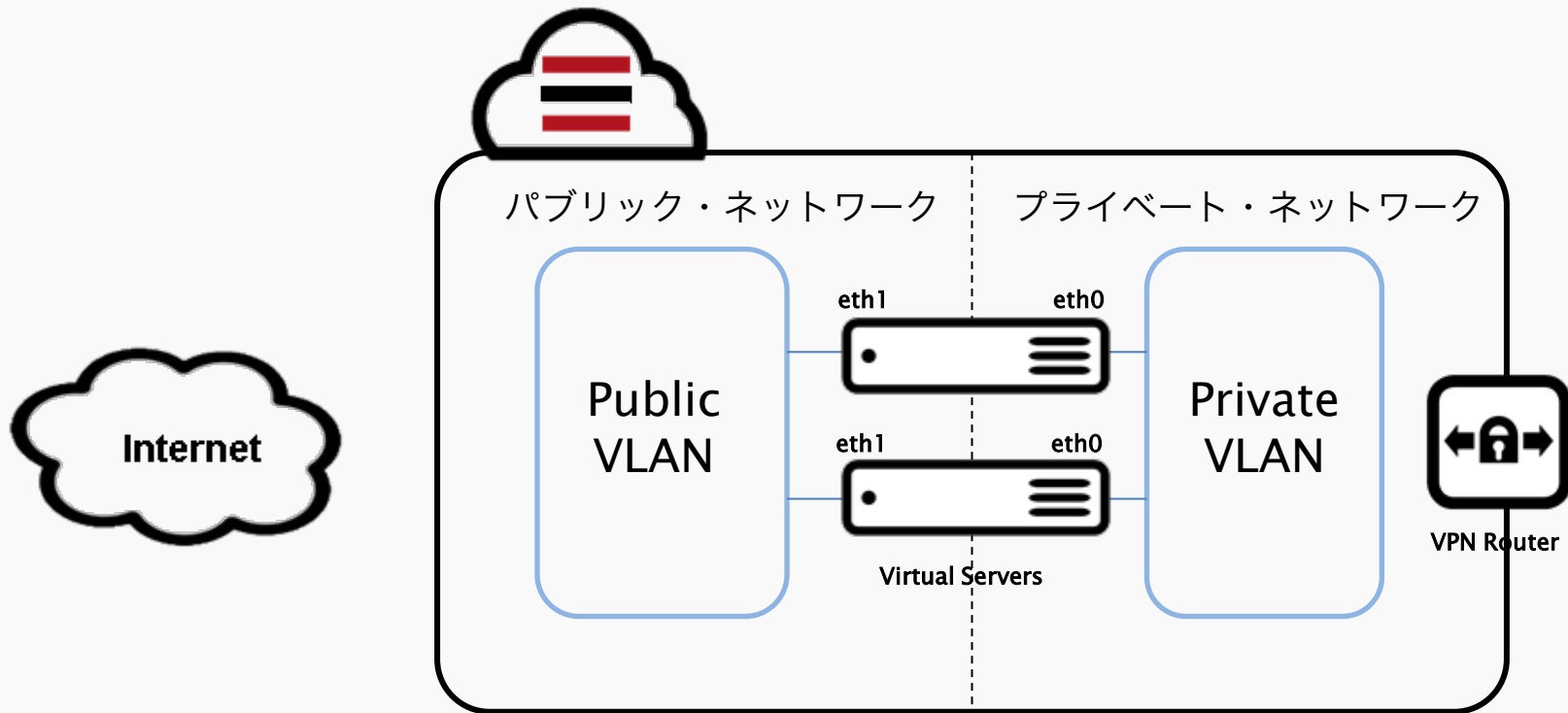


パブリック・ネットワーク

プライベート・ネットワーク



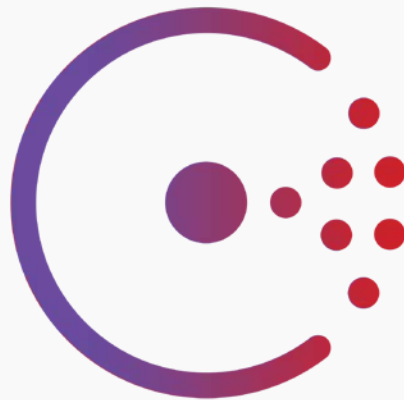
Virtual Servers





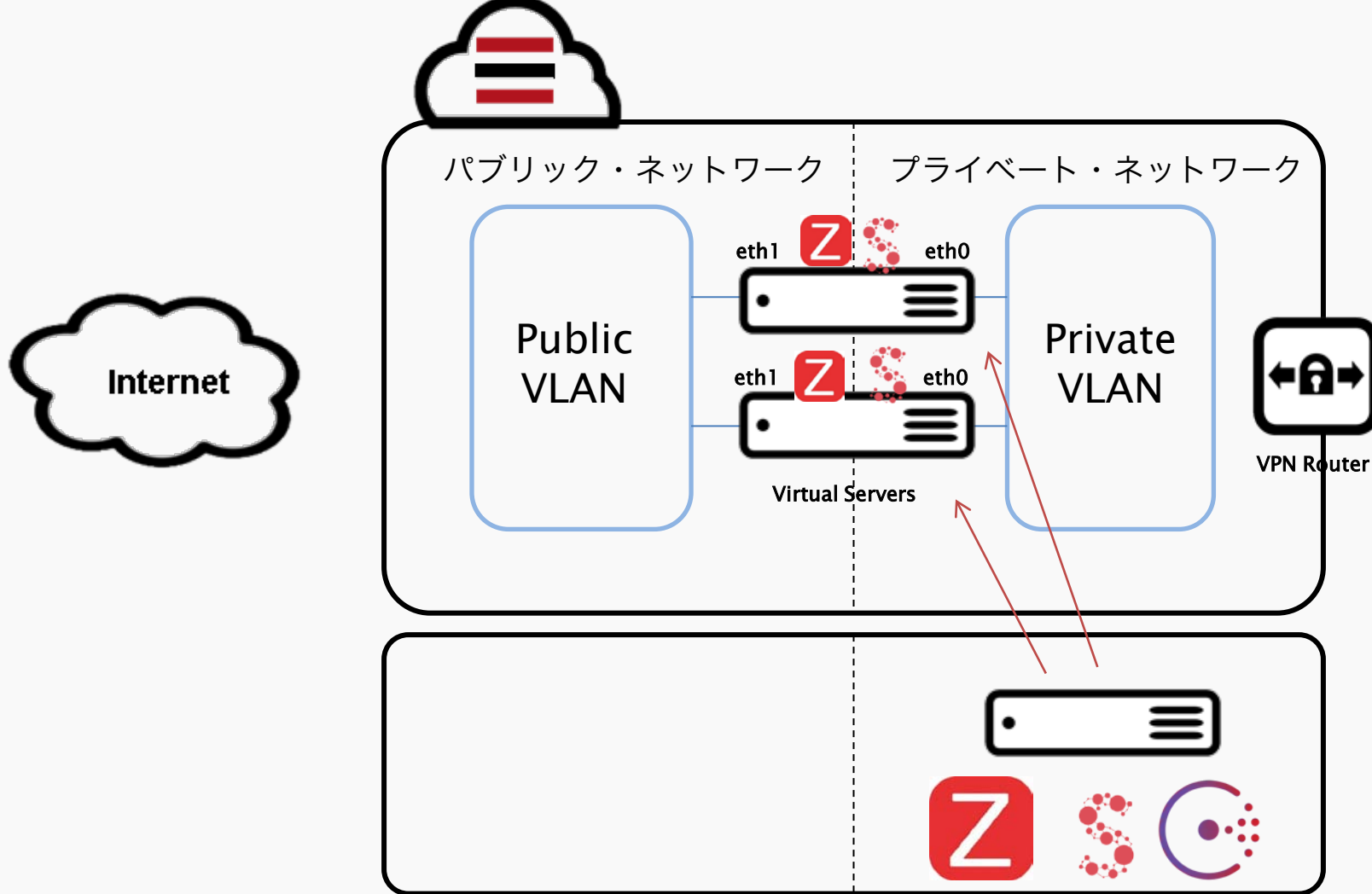
Serf

<http://www.serfdom.io/>

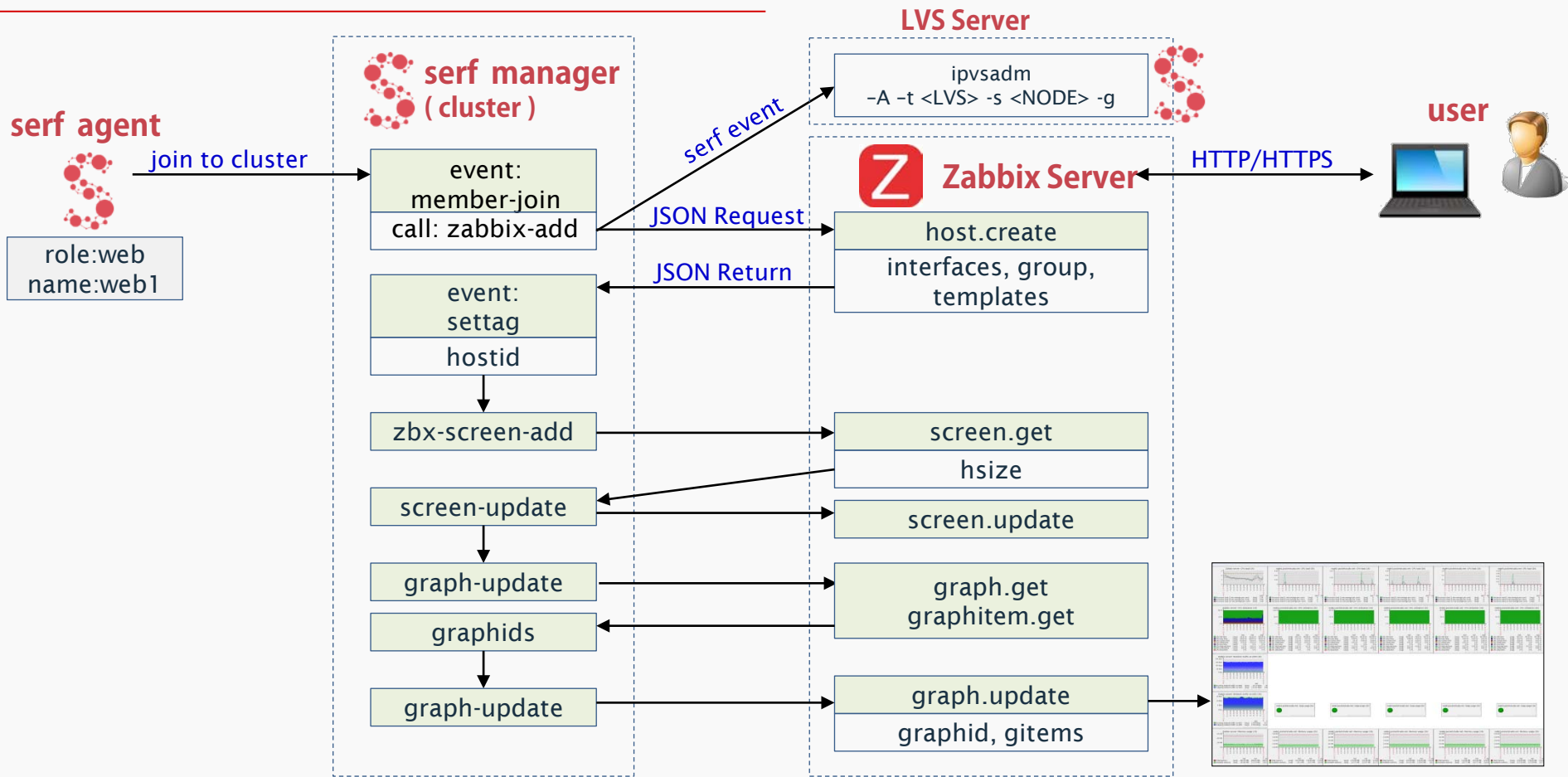


Consul

<http://www.consul.io/>



Workflow orchestration





serf manager

```
$ serf agent -iface=eth1 -discover=serf -log-level=debug ¥  
--event-handler=/opt/serf-lvs/ADP.pl
```

イベントが発生する度に、この Perl スクリプトを実行。
イベント毎に、ZABBIX Server に対する JSON リクエストや
LVS (ipvsadm) を制御します。このとき、処理対象の role
も判断材料に用います。



serf node

```
$ serf agent -iface=eth1 -discover=serf -log-level=debug ¥  
-event-handler=user:settag=/opt/serf/changetag.pl -tag role=web
```

serf のタグで 'web' や 'db' など、サーバの役割を明示。
web であれば、HTTP の監視を行い、LVS に追加。
db であれば、MySQL の監視を行うように区別します。

ZABBIXヘルプ | サポート | プリント | プロファイル | ログアウト

監視データ | インベントリ | レポート | 設定 | 管理

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

検索

履歴: ユーザープロフィール » Custom graphs » Dashboard » Configuration of hosts » User profile

ホストの設定

ホストの作成 | インポート

ホスト

グループ

すべて

Displaying 1 to 3 of 3 found

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

Web

インターフェース

テンプレート

ステータス

エージェントの状態

node1.pocketstudio.net

アプリケーション (14)

アイテム (37)

トリガー (20)

グラフ (9)

ディスカバリ (2)

Web (1)

192.168.39.11: 10050

Template App HTTP Service (Template Web Response), Template App SSH Service, Template ICMP Ping, Template OS Linux (Template App Zabbix Agent)

有効

Overview Layer

アプリケーション (0)

アイテム (0)

トリガー (0)

グラフ (0)

ディスカバリ (0)

Web (0)

127.0.0.1: 10050

-

有効

Zabbix server

アプリケーション (11)

アイテム (77)

トリガー (46)

グラフ (22)

ディスカバリ (2)

Web (0)

127.0.0.1: 10050

Template App Zabbix Server, Template OS Linux (Template App Zabbix Agent)

有効

選択をエクスポート

実行 (0)

Zabbix 2.2.2 Copyright 2001-2014 by Zabbix SIA

ユーザー 'Admin' でログイン

serf agent を起動するだけで、自動的にホスト登録と監視を開始。ZABBIX 画面に一切ログインすることなく設定を行えます（Serfと連携し、IP アドレスの情報を自動的に渡しています）。

ホスト

グループ すべて

Displaying 1 to 5 of 5 found

フィルター

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	Web	インターフェース	テンプレート	ステータス	エージェントの状態
<input type="checkbox"/> node1.pocketstudio.net	アプリケーション (14)	アイテム (37)	トリガー (20)	グラフ (9)	ディスカバリ (2)	Web (1)	192.168.39.11: 10050	Template App HTTP Service (Template Web Response), Template App SSH Service, Template ICMP Ping, Template OS Linux (Template App Zabbix Agent)	有効	
<input type="checkbox"/> node2.pocketstudio.net	アプリケーション (14)	アイテム (37)	トリガー (20)	グラフ (9)	ディスカバリ (2)	Web (1)	192.168.39.12: 10050	Template App HTTP Service (Template Web Response), Template App SSH Service, Template ICMP Ping, Template OS Linux (Template App Zabbix Agent)	有効	
<input type="checkbox"/> node3.pocketstudio.net	アプリケーション (13)	アイテム (50)	トリガー (20)	グラフ (11)	ディスカバリ (2)	Web (0)	192.168.39.13: 10050	Template App MySQL, Template App SSH Service, Template ICMP Ping, Template OS Linux (Template App Zabbix Agent)	有効	
<input type="checkbox"/> Overview Layer	アプリケーション (0)	アイテム (0)	トリガー (0)	グラフ (0)	ディスカバリ (0)	Web (0)	127.0.0.1: 10050	-	有効	
<input type="checkbox"/> Zabbix server	アプリケーション (11)	アイテム (77)	トリガー (46)	グラフ (22)	ディスカバリ (2)	Web (0)	127.0.0.1: 10050	Template App Zabbix Server, Template OS Linux (Template App Zabbix Agent)	有効	

HTTP
Template

MySQL
Template

Zoom: 1h 2h 3h 6h 12h 1d 7d 14d 1m 3m 6m 1y All

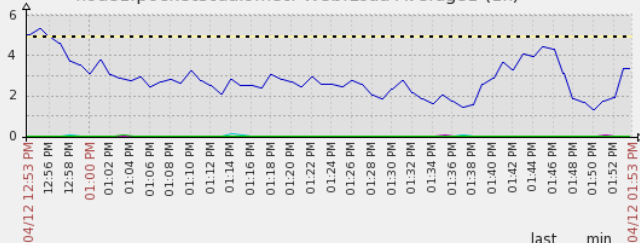
Apr 12th, 2014 01:53 PM - Apr 12th, 2014 02:53 PM (now!)



<< 1y 6m 1m 7d 1d 12h 1h | 1h 12h 1d 7d 1m 6m 1y >>

1h (fixed)

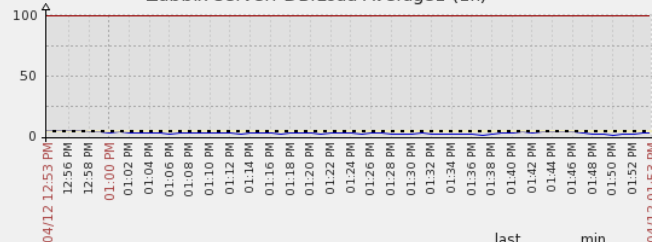
node1.pocketstudio.net: Web:Load Average1 (1h)



node4.pocketstudio.net: Processor load (1 min average per core)	[avg]	0	0	avg
node3.pocketstudio.net: Processor load (1 min average per core)	[avg]	0	0	0.00
node2.pocketstudio.net: Processor load (1 min average per core)	[avg]	0	0	0.00
node1.pocketstudio.net: Processor load (1 min average per core)	[avg]	0	0	0.00
Zabbix server: Processor load (1 min average per core)	[avg]	3.34	1.35	2

Data from history. Generated in 2.25 sec

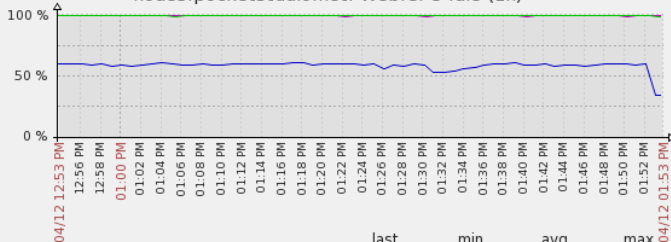
Zabbix server: DB:Load Average1 (1h)



node5.pocketstudio.net: CPU idle time	[avg]	99.55 %	99.55 %	99.78 %
Zabbix server: Processor load (1 min average per core)	[avg]	3.34	1.35	2

Data from history. Generated in 1.11 sec

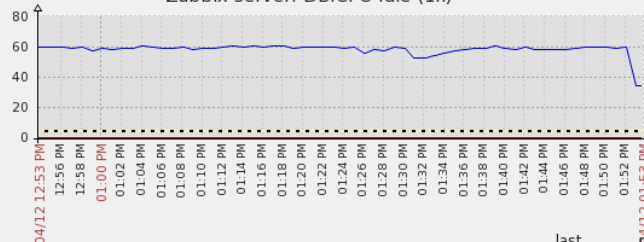
node3.pocketstudio.net: Web:CPU Idle (1h)



node4.pocketstudio.net: CPU idle time	[avg]	99.73 %	99.72 %	99.78 %	99.85 %
node3.pocketstudio.net: CPU idle time	[avg]	99.67 %	99.6 %	99.71 %	99.77 %
node2.pocketstudio.net: CPU idle time	[avg]	99.63 %	99.36 %	99.69 %	99.78 %
node1.pocketstudio.net: CPU idle time	[avg]	99.5 %	99.48 %	99.65 %	99.75 %
Zabbix server: CPU idle time	[avg]	34.21 %	34.21 %	58.65 %	60.95 %

Data from history. Generated in 1.98 sec

Zabbix server: DB:CPU Idle (1h)



node5.pocketstudio.net: Processor load (1 min average per core)	[avg]	0	0	0
Zabbix server: CPU idle time	[avg]	34.21 %	34.21 %	34.21 %

Data from history. Generated in 0.94 sec

■ Serf エージェント依存

- ➡ サービスレベルの監視ができない
- ➡ roleを都度明示する必要がある

■ Consul を使った監視自動化を模索中

- ➡ consul エージェントが稼働中のサービスを判別
- ➡ サービスの稼働状況に応じて、テンプレート自動適用
- ➡ メンテナンス・フリーの監視環境

SoftLayerのクラウド環境を
Zabbixで効率的に監視するには

Zabbix as framework and Effective Monitoring

5 実践SoftLayer

Practical operation

■ プライベートVLANの活用

➡ 安全な環境を作れる

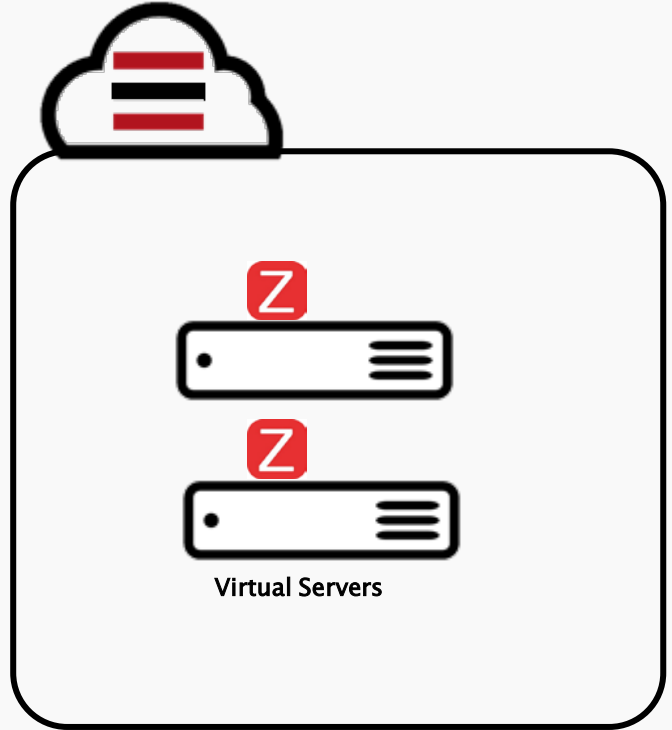
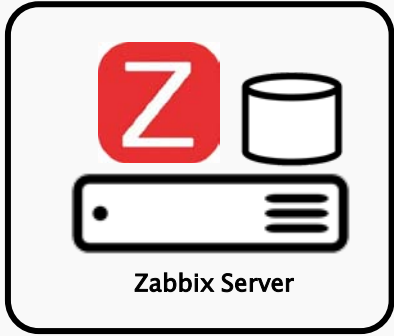
- VPN はアカウント毎に接続先のサブネットを制限できる
- アカウント数の制限なし

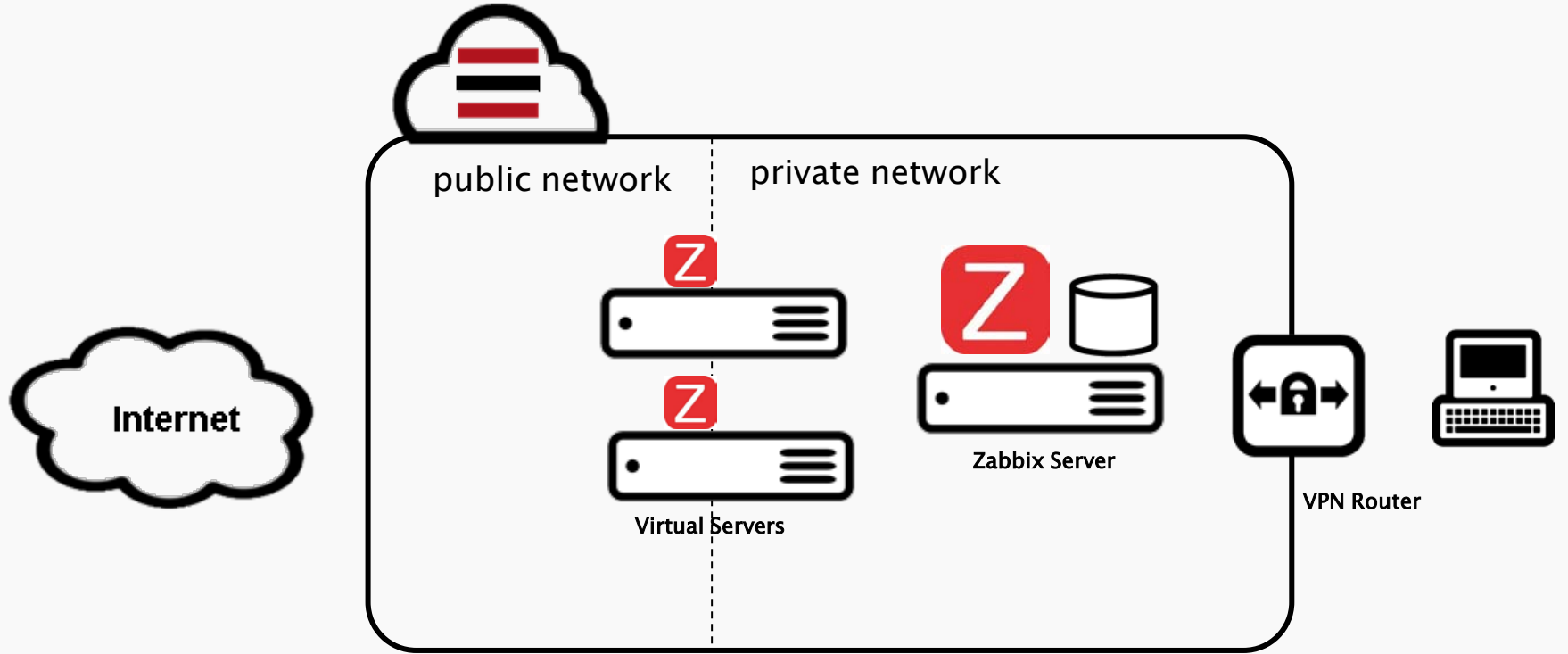
➡ エージェントのポートをパブリック側に公開不要

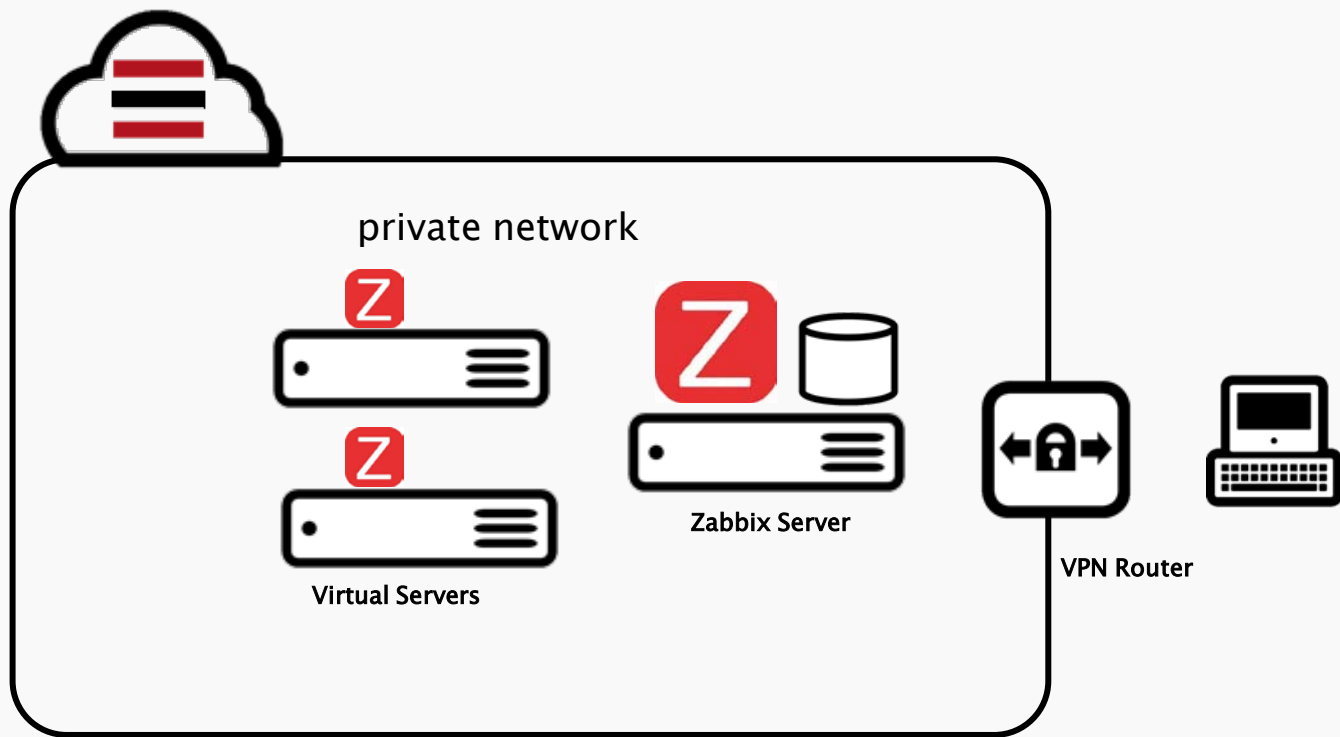
- 内部からの監視を特に意識しなくても出来る

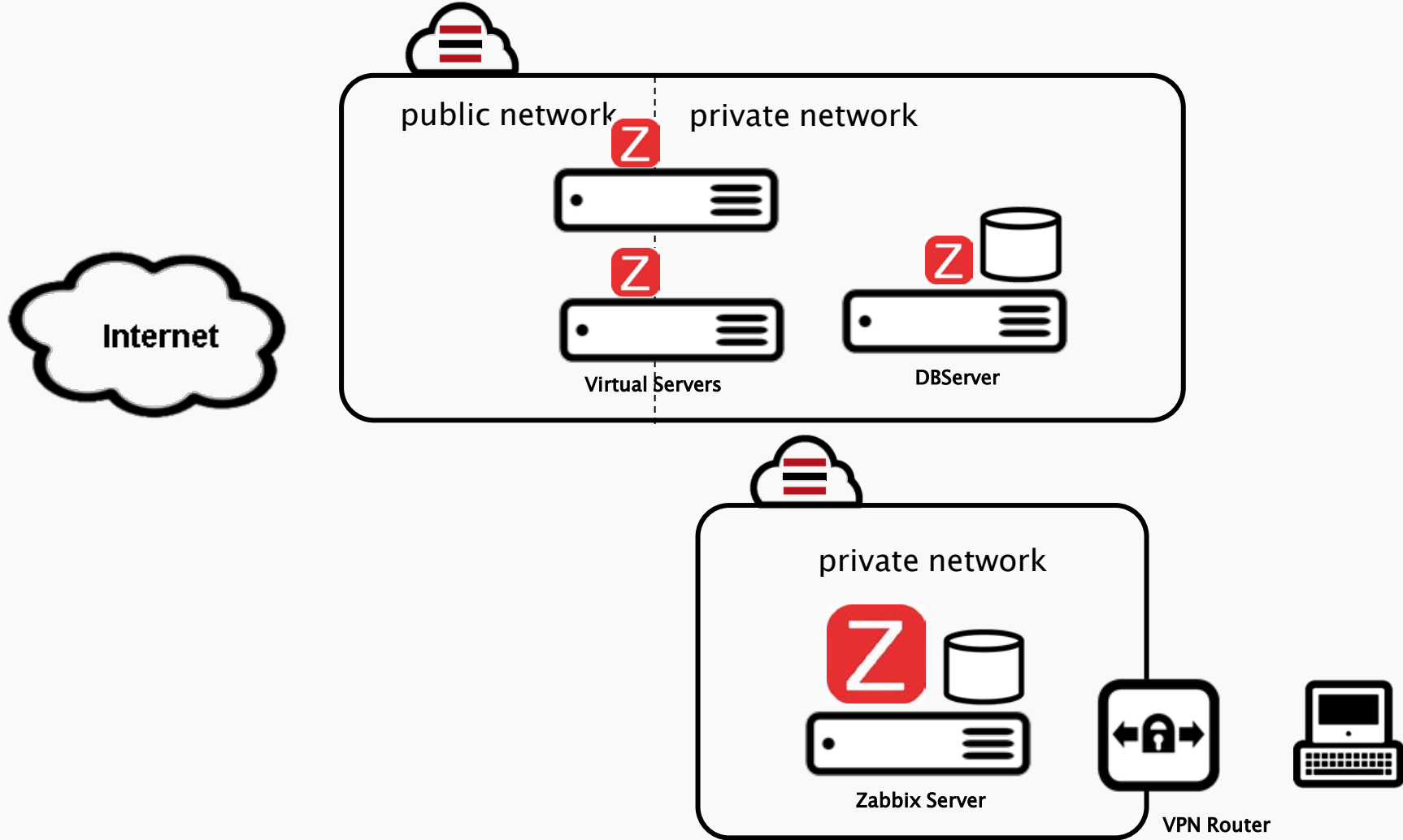
➡ プライベート・ネットワーク内の課金なし

- 通信量無料なので、複数ネットワークを内部から監視可能
- 接続できない環境とは、zabbix-proxy の活用









SoftLayerとZabbixの連携

■ 自動監視

- ➡ サーバ追加・削除の連動
- ➡ roleとtagに応じて、グルーピング
- ➡ オートスケール時に自動適用

■ API連携

- ➡ 利用料金計算
- ➡ チケット検出

Provisioning Script と Metadata

■ Provisioning Script

- ➡ サーバ起動時に自動実行できるスクリプト
 - zabbix agent の自動セットアップ
 - serf, consul の自動セットアップ

■ Metadata

- ➡ Provisioning Script に変数を指定できる
 - Zabbix Server の IP アドレス指定
 - serf, consul の cluster join 先指定

APIを使ってみたい！

■ 3種類のプロトコルを選ぶ

➡ SOAP

- SoftLayer API に直接つながるので最速
- 推奨言語：C#、Perl、PHP、VisualBasic.NET、等

➡ XML-RPC

- 多くの言語に対応するが、主にSOAPしない言語向け
- 推奨言語：PHP、Python

➡ REST

- HTTPプロトコルで簡単にデータを取得できる
- 推奨言語：Ruby等、HTTPプロトコルを扱えるもの



Getting Started | SoftLayer
Development Network
[http://sldn.softlayer.com/article/
Getting-Started](http://sldn.softlayer.com/article/Getting-Started)

■ RESTがおすすめの理由

- ➡ curlを使って簡単に確認できる
 - 冴えるシェル芸
- ➡ 出力結果を選択でき、用途に応じて加工しやすい
 - テキスト（文字列）
 - XML形式
 - JSON形式

API Key

■ API Keyの特長

➡ アカウント単位なので、権限設定ができる

■ 確認方法

➡ 【 Account 】 → 【 Users 】

The screenshot shows the SOFTLAYER user management interface. The top navigation bar includes links for Support, Devices, Storage, Network, Security, Services, and Account. The Account menu is expanded, showing a dropdown with a checkmark. The main content area displays the 'Users' page, showing a list of users. A modal dialog box is open, displaying the 'API Key for SL302402' as 'd0e534cf7a46077cf94663c230eed1e58d48d75e21fa4048c8bfe300bb8b39c0'. The dialog has an 'OK' button. The user list table below shows columns for Last Name, First Name, Username, Status, and Actions. The first user is 'Zembutsu Masahito' with status 'Active' and a 'Manage' link. The second user is 'Zembutsu2 Masahito2' with status 'Active' and a 'Manage' link. The third user is 'Zembutsu' with status 'Active' and a 'Manage' link. The fourth user is 'Zembutsu' with status 'Active' and a 'Manage' link. The fifth user is 'Zembutsu2' with status 'Active' and a 'Manage' link. The sixth user is 'Zembutsu' with status 'Active' and a 'Manage' link. The seventh user is 'Zembutsu' with status 'Active' and a 'Manage' link. The eighth user is 'Zembutsu' with status 'Active' and a 'Manage' link. The ninth user is 'Zembutsu' with status 'Active' and a 'Manage' link. The tenth user is 'Zembutsu' with status 'Active' and a 'Manage' link. The eleventh user is 'Zembutsu' with status 'Active' and a 'Manage' link. The twelfth user is 'Zembutsu' with status 'Active' and a 'Manage' link. The thirteenth user is 'Zembutsu' with status 'Active' and a 'Manage' link. The fourteenth user is 'Zembutsu' with status 'Active' and a 'Manage' link. The fifteenth user is 'Zembutsu' with status 'Active' and a 'Manage' link. The sixteenth user is 'Zembutsu' with status 'Active' and a 'Manage' link. The seventeenth user is 'Zembutsu' with status 'Active' and a 'Manage' link. The eighteenth user is 'Zembutsu' with status 'Active' and a 'Manage' link. The nineteenth user is 'Zembutsu' with status 'Active' and a 'Manage' link. The twentieth user is 'Zembutsu' with status 'Active' and a 'Manage' link. The twenty-first user is 'Zembutsu' with status 'Active' and a 'Manage' link. The twenty-second user is 'Zembutsu' with status 'Active' and a 'Manage' link. The twenty-third user is 'Zembutsu' with status 'Active' and a 'Manage' link. The twenty-fourth user is 'Zembutsu' with status 'Active' and a 'Manage' link. The twenty-fifth user is 'Zembutsu' with status 'Active' and a 'Manage' link. The twenty-sixth user is 'Zembutsu' with status 'Active' and a 'Manage' link. The twenty-seventh user is 'Zembutsu' with status 'Active' and a 'Manage' link. The twenty-eighth user is 'Zembutsu' with status 'Active' and a 'Manage' link. The twenty-ninth user is 'Zembutsu' with status 'Active' and a 'Manage' link. The thirtieth user is 'Zembutsu' with status 'Active' and a 'Manage' link. The thirty-first user is 'Zembutsu' with status 'Active' and a 'Manage' link. The thirty-second user is 'Zembutsu' with status 'Active' and a 'Manage' link. The thirty-third user is 'Zembutsu' with status 'Active' and a 'Manage' link. The thirty-fourth user is 'Zembutsu' with status 'Active' and a 'Manage' link. The thirty-fifth user is 'Zembutsu' with status 'Active' and a 'Manage' link. The thirty-sixth user is 'Zembutsu' with status 'Active' and a 'Manage' link. The thirty-seventh user is 'Zembutsu' with status 'Active' and a 'Manage' link. The thirty-eighth user is 'Zembutsu' with status 'Active' and a 'Manage' link. The thirty-ninth user is 'Zembutsu' with status 'Active' and a 'Manage' link. The fortieth user is 'Zembutsu' with status 'Active' and a 'Manage' link. The forty-first user is 'Zembutsu' with status 'Active' and a 'Manage' link. The forty-second user is 'Zembutsu' with status 'Active' and a 'Manage' link. The forty-third user is 'Zembutsu' with status 'Active' and a 'Manage' link. The forty-fourth user is 'Zembutsu' with status 'Active' and a 'Manage' link. The forty-fifth user is 'Zembutsu' with status 'Active' and a 'Manage' link. The forty-sixth user is 'Zembutsu' with status 'Active' and a 'Manage' link. The forty-seventh user is 'Zembutsu' with status 'Active' and a 'Manage' link. The forty-eighth user is 'Zembutsu' with status 'Active' and a 'Manage' link. The forty-ninth user is 'Zembutsu' with status 'Active' and a 'Manage' link. The fiftieth user is 'Zembutsu' with status 'Active' and a 'Manage' link. The fifty-first user is 'Zembutsu' with status 'Active' and a 'Manage' link. The fifty-second user is 'Zembutsu' with status 'Active' and a 'Manage' link. The fifty-third user is 'Zembutsu' with status 'Active' and a 'Manage' link. The fifty-fourth user is 'Zembutsu' with status 'Active' and a 'Manage' link. The fifty-fifth user is 'Zembutsu' with status 'Active' and a 'Manage' link. The fifty-sixth user is 'Zembutsu' with status 'Active' and a 'Manage' link. The fifty-seventh user is 'Zembutsu' with status 'Active' and a 'Manage' link. The fifty-eighth user is 'Zembutsu' with status 'Active' and a 'Manage' link. The fifty-ninth user is 'Zembutsu' with status 'Active' and a 'Manage' link. The sixtieth user is 'Zembutsu' with status 'Active' and a 'Manage' link. The sixty-first user is 'Zembutsu' with status 'Active' and a 'Manage' link. The sixty-second user is 'Zembutsu' with status 'Active' and a 'Manage' link. The sixty-third user is 'Zembutsu' with status 'Active' and a 'Manage' link. The sixty-fourth user is 'Zembutsu' with status 'Active' and a 'Manage' link. The sixty-fifth user is 'Zembutsu' with status 'Active' and a 'Manage' link. The sixty-sixth user is 'Zembutsu' with status 'Active' and a 'Manage' link. The sixty-seventh user is 'Zembutsu' with status 'Active' and a 'Manage' link. The sixty-eighth user is 'Zembutsu' with status 'Active' and a 'Manage' link. The sixty-ninth user is 'Zembutsu' with status 'Active' and a 'Manage' link. The seventieth user is 'Zembutsu' with status 'Active' and a 'Manage' link. The seventy-first user is 'Zembutsu' with status 'Active' and a 'Manage' link. The seventy-second user is 'Zembutsu' with status 'Active' and a 'Manage' link. The seventy-third user is 'Zembutsu' with status 'Active' and a 'Manage' link. The seventy-fourth user is 'Zembutsu' with status 'Active' and a 'Manage' link. The seventy-fifth user is 'Zembutsu' with status 'Active' and a 'Manage' link. The seventy-sixth user is 'Zembutsu' with status 'Active' and a 'Manage' link. The seventy-seventh user is 'Zembutsu' with status 'Active' and a 'Manage' link. The seventy-eighth user is 'Zembutsu' with status 'Active' and a 'Manage' link. The seventy-ninth user is 'Zembutsu' with status 'Active' and a 'Manage' link. The eightieth user is 'Zembutsu' with status 'Active' and a 'Manage' link. The eighty-first user is 'Zembutsu' with status 'Active' and a 'Manage' link. The eighty-second user is 'Zembutsu' with status 'Active' and a 'Manage' link. The eighty-third user is 'Zembutsu' with status 'Active' and a 'Manage' link. The eighty-fourth user is 'Zembutsu' with status 'Active' and a 'Manage' link. The eighty-fifth user is 'Zembutsu' with status 'Active' and a 'Manage' link. The eighty-sixth user is 'Zembutsu' with status 'Active' and a 'Manage' link. The eighty-seventh user is 'Zembutsu' with status 'Active' and a 'Manage' link. The eighty-eighth user is 'Zembutsu' with status 'Active' and a 'Manage' link. The eighty-ninth user is 'Zembutsu' with status 'Active' and a 'Manage' link. The ninetieth user is 'Zembutsu' with status 'Active' and a 'Manage' link. The ninety-first user is 'Zembutsu' with status 'Active' and a 'Manage' link. The ninety-second user is 'Zembutsu' with status 'Active' and a 'Manage' link. The ninety-third user is 'Zembutsu' with status 'Active' and a 'Manage' link. The ninety-fourth user is 'Zembutsu' with status 'Active' and a 'Manage' link. The ninety-fifth user is 'Zembutsu' with status 'Active' and a 'Manage' link. The ninety-sixth user is 'Zembutsu' with status 'Active' and a 'Manage' link. The ninety-seventh user is 'Zembutsu' with status 'Active' and a 'Manage' link. The ninety-eighth user is 'Zembutsu' with status 'Active' and a 'Manage' link. The ninety-ninth user is 'Zembutsu' with status 'Active' and a 'Manage' link. The hundredth user is 'Zembutsu' with status 'Active' and a 'Manage' link.

SOFTLAYER[®]
an IBM Company

Logged in as: Masahito Zembutsu | Creationline Inc. (302402) | [Log Out](#)

Support ▾ | Devices ▾ | Storage ▾ | Network ▾ | Security ▾ | Services ▾ | **Account ▾** |

Search Portal

Users [Add User](#)

Viewing 1 to 3 of 3 [Active Users](#) [Filter](#)

Last Name	First Name	Username	Status	Manage	None	View	Actions
Zembutsu	Masahito						View
Zembutsu	Masahito						View
Zembutsu2	Masahito2	sl302402-zem2	Active	Manage	None	View	Actions

ケース；利用料金を知りたい！

■ “クラウド破産”を防ぐには？

☞ クラウド破産とは、クラウドの利用料金が従量課金のため、意図しない支出を強いられる現象のこと。

Amazon Web Services Invoice	
Email or talk to us about your AWS account or bill, visit aws.amazon.com/contact-us/	
Invoice Summary	
Invoice Number:	
Invoice Date:	May 3 , 2012
TOTAL AMOUNT DUE ON May 3, 2012	\$248.74

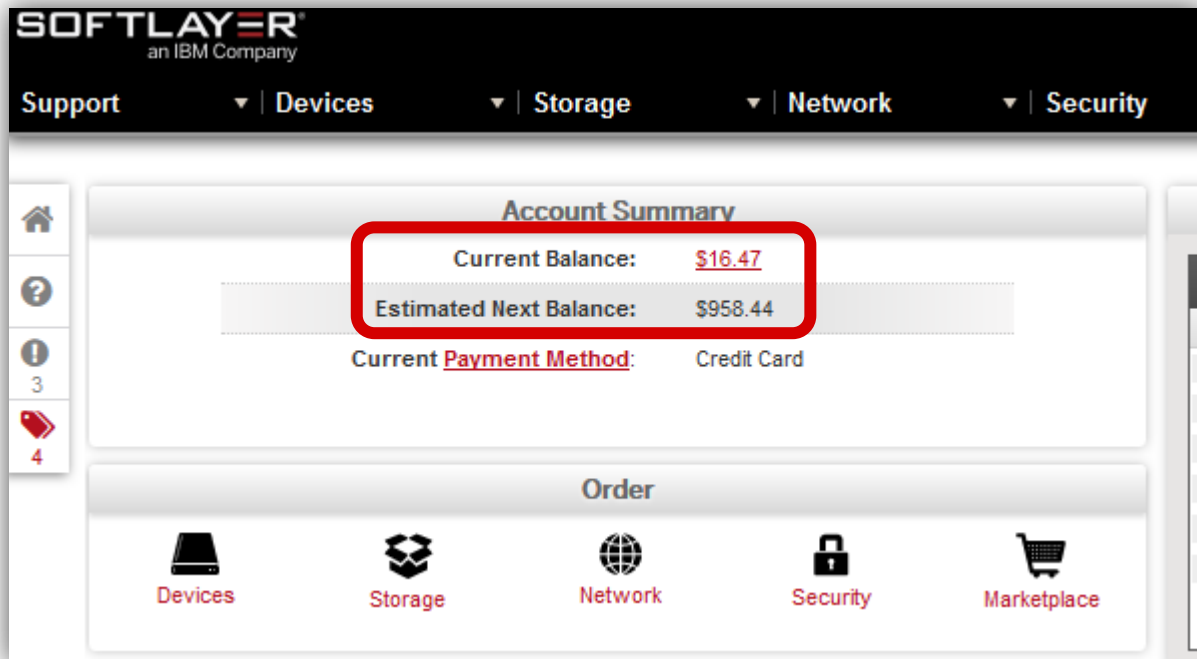
AWS利用料金グラフ化Muninプラグインを書いてみた（ ^ ω ^ ） | Pocketstudio.jp log3
<http://pocketstudio.jp/log3/2012/05/21/aws-estimated-charge-munin-plugin/>

ケース；利用料金を知りたい！

■ SoftLayerの利用料金

➡ 時間単位(hourly) + 月額(monthly)

👉 ポータルの【Account】 →
【Billing】 → 【Invoices】



The screenshot shows the SoftLayer account management interface. At the top, the header includes the SoftLayer logo (an IBM Company) and navigation tabs for Support, Devices, Storage, Network, and Security. On the left, a sidebar contains icons for Home, Help, Alerts (3), and Shopping Cart (4). The main content area is titled 'Account Summary' and contains a table with the following data:

Account Summary	
Current Balance:	<u>\$16.47</u>
Estimated Next Balance:	\$958.44
Current <u>Payment Method</u> :	Credit Card

Below the Account Summary section, there is an 'Order' section with five icons representing different services: Devices, Storage, Network, Security, and Marketplace.

料金を取得するには？

■ API 鍵のファイルを用意

➡ 例； "api.key"

```
user=sl342860-zembutsu:685ce01489f71fbea44...b69841175517e3
```

➡ curl で実行

```
$ curl -s -K ./api.key 'https://api.softlayer.com/rest/v3/SoftLayer_Account/getBalance'  
"16.47"
```

```
$ curl -s -K ./api.key ¥  
'https://https://api.softlayer.com/rest/v3/SoftLayer_Account/getNextInvoiceTotalRecurringAmount'  
"746.18"
```

■ Zabbixにデータ送信

➡ "zabbix_sender"

```
$ zabbix_sender -z <zabbix_host> -s <hostname> -k <keyname> -o <data>
```

今後の拡張アイデア 1

■ チケットをZabbixで監視

➡ https://api.softlayer.com/rest/v3/SoftLayer_Account/getOpenTickets

```
[
  {
    "status": {
      "name": "Open",
      "id": 1001
    },
    "userEditableFlag": false,
    "title": "Important Changes Regarding your Invoice - All - 10/01/2014",
    "subjectId": null,
    "statusId": 1001,
    "serviceProviderResourceId": 12654290,
    "serviceProviderId": 1,
    "serverAdministrationRefundInvoiceId": null,
    "serverAdministrationFlag": 0,
    "lastEditDate": "2014-09-24T13:27:38+09:00",
    "id": 12654290,
    "groupId": 1018,
    "createDate": "2014-09-24T13:27:38+09:00",
    "changeOwnerFlag": false,
    "billableFlag": null,
    "assignedUserId": 211088,
    "accountId": 342860,
    "lastEditType": "AUTO",
```

今後の拡張アイデア 1

■ チケットをZabbixで監視

➡ https://api.softlayer.com/rest/v3/SoftLayer_Account/getActiveAlarms

```
[
  {
    "severity": "critical",
    "robotId": 389520,
    "message": "Robot 342860-C-00081-z-web01 is inactive",
    "createDate": "2014-11-12T21:25:14+09:00",
    "closedDate": null,
    "alarmId": "RN15777924-44221",
    "agentId": null,
    "accountId": 342860
  },
  {
    "severity": "critical",
    "robotId": 385492,
    "message": "Robot 342860-C-00080-z-web01 is inactive",
    "createDate": "2014-11-12T21:25:14+09:00",
    "closedDate": null,
    "alarmId": "RN15777924-44256",
    "agentId": null,
    "accountId": 342860
  }
]
```

今後の拡張アイディア 2

■ 障害情報の一元管理

➡ <http://lg.softlayer.com/>

	ATL	CHI	DAL	DEN	HOU	LAX	MIA	NYC	SEA	SJC	TOR	WDC	AMS	FRA	LON	PAR	HKG	MEL	SNG	TOK
ATL	0	31	20	35	28	49	13	18	60	59	33	13	91	97	88	92	200	199	225	165
CHI	31	0	20	25	28	55	43	22	45	52	11	19	93	105	89	96	196	204	220	152
DAL	20	20	0	15	8	32	30	48	41	39	31	35	111	119	109	115	191	181	203	138
DEN	35	25	15	0	24	34	44	47	27	28	36	43	120	127	115	120	176	184	194	127
HOU	28	28	8	24	0	38	24	45	49	42	37	42	119	124	115	124	189	182	209	146
LAX	49	55	32	34	38	0	60	68	25	8	67	60	139	144	143	145	157	149	173	101
MIA	13	43	30	44	24	60	0	32	73	67	45	26	105	109	100	104	215	206	230	175
NYC	18	22	48	47	45	68	32	0	66	74	15	6	74	78	69	74	217	221	254	173
SEA	60	45	41	27	49	25	73	66	0	18	56	63	139	147	134	139	162	174	198	117
SJC	59	52	39	28	42	8	67	74	18	0	63	71	146	151	142	148	145	157	180	106
TOR	33	11	31	36	37	67	45	15	56	63	0	21	88	96	83	89	212	215	230	162
WDC	13	19	35	43	42	60	26	6	63	71	21	0	80	87	75	81	220	216	244	165
AMS	91	93	111	120	119	139	105	74	139	146	88	80	0	6	7	15	300	294	334	245
FRA	97	105	119	127	124	144	109	78	147	151	96	87	6	0	11	9	304	298	324	257
LON	88	89	109	115	115	143	100	69	134	142	83	75	7	11	0	7	291	289	322	235
PAR	92	96	115	120	124	145	104	74	139	148	89	81	15	9	7	0	291	293	315	246
HKG	200	196	191	176	189	157	215	217	162	145	212	220	300	304	291	291	0	129	31	50
MEL	199	204	181	184	182	149	206	221	174	157	215	216	294	298	289	293	129	0	158	127
SNG	225	220	203	194	209	173	230	254	198	180	230	244	334	324	322	315	31	158	0	68
TOK	165	152	138	127	146	101	175	173	117	106	162	165	245	257	235	246	50	127	68	0

SoftLayer IP Backbone - Datacenter / Network Information

■ <http://www.softlayer.com/data-centers>

TOK - SJC			
Latency - 5 Minute Average			
	Current	Warning	Critical
Latency:	106ms	132ms	158ms
Loss:	0%	2%	4%

今後の拡張アイデア 2

■ 障害情報の一元管理

➡ @SoftLayerNotify



今後の拡張アイデア 3

■ 監視設定の完全自動化

➡ Consul, Serf 等と連動

■ アラート発生時の自動対応

➡ Consul 等のイベントハンドラ自動実行

■ 他のツールとの連携

➡ Slack

➡ hubot

➡ hue

■ docker監視対応

➡ コンテナ環境の監視に応用



<http://www.slideshare.net/zembutsu/re-is-the-order-an-automation-2nd>

**監視を手動でしてるんだが、もう俺は限界かもしれない
重要な点は、設定時間短縮・人的作業事故を回避**

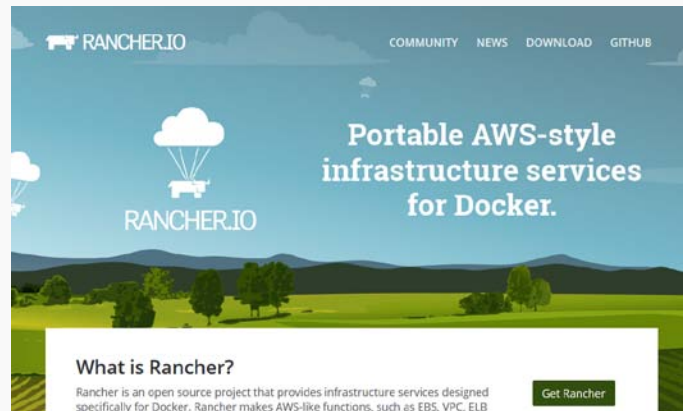
クラウドは対応できた

■ 今度はコンテナだ！

➡ Google Container Engine



➡ Amazon EC2 Container Service



この先生きのこりたい！

運用を続けるためには、イカの質問に答える必要があります。

It is necessary to answer the following questions to start OPERATION.

ニンゲンヤメマスカ?

Do you resign as human being?

YES

はい

はい

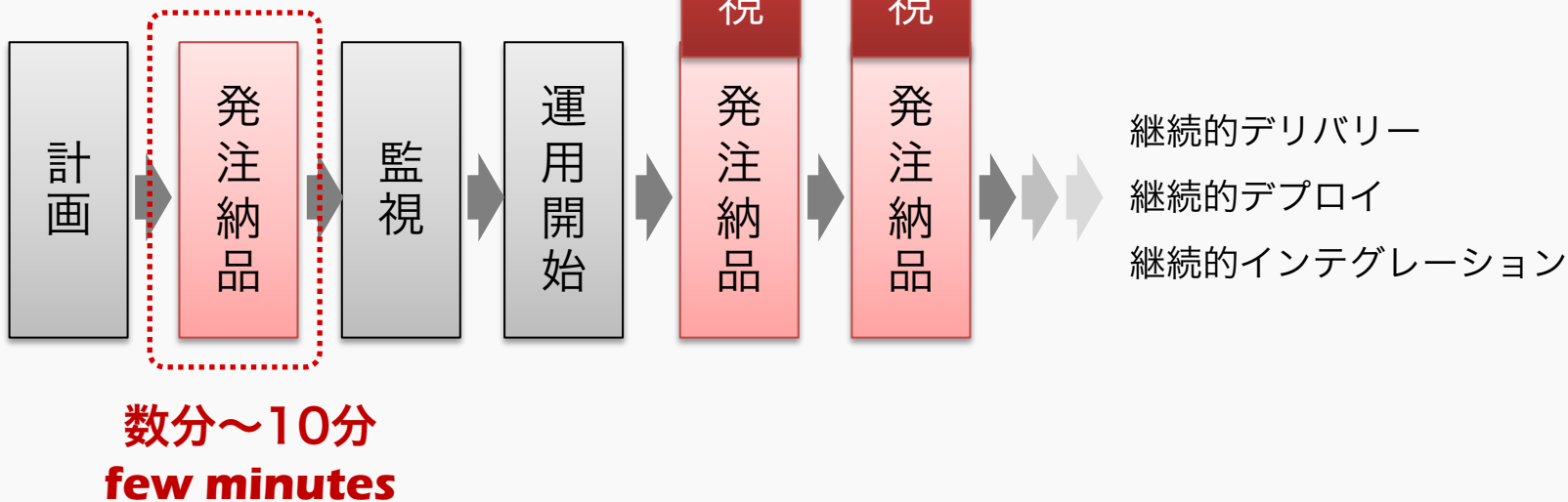
YES

Xen



やったぜ！Zabbix

クラウド・コンピューティング基盤 Infrastructure as a Service



SoftLayerのクラウド環境を
Zabbixで効率的に監視するには

Zabbix as framework and Effective Monitoring

Zabbix as a **framework**

ソフトウェアフレームワーク(英: *software framework*)とは、プログラミングにおいて、一般的な機能をもつ共通コードをユーザーが選択的に上書きしたり特化させたりすることで、ある特定の機能をもたせようとする抽象概念のことである。単にフレームワークとも呼ばれる。

SoftLayerのクラウド環境を
Zabbixで効率的に監視するには

Zabbix as framework and Effective Monitoring

MSP業務にZabbixを**何故導入**したのか？

どのように**拡張**していくのか？ 鍵は **API**

SoftLayerの情報

■ 公式

➡ <http://www.softlayer.com/>

■ 日本語情報

➡ SoftLayerユーザ会

- <http://jslug.jp/>

➡ ソフトレイヤー活用技術資料

- <https://gg-web.jp/document/ConfigGuide/>

