GMOインターネットのOpenStack public cloud “ConoHa”におけるオープンソース監視ツール “Hatohol” と”zabbix”の適用について

How to use opensource monitoring tools Hatohol and zabbix on GMO Ineternet using OpenStack for Public Cloud

Slide URL
http://www.slideshare.net/chroum/miracle-linux-seminer-hatohol-and-conoha

ConoHa public cloud
https://www.conoha.jp/

Miracle Linux ZBX/Hatohol セミナー

Feb 17, 2017

Naoto Gohko (@naoto_gohko)

GMO Internet, Inc.

内容(アジェンダ)

- 0) GMOインターネットのOpenStack public Cloudサービスについて
- 1) 運用管理とはなんぞや
- 2) これまで運用管理に使っていたツールについて
- 3) ConoHaをマルチロケーションで運用するにあたって再検討した最近の運用監視ツールについて
- 4) Hatohol + zabbixを選択した理由について
- 5) Hatohol + zabbix構成について

- 6) (参考資料) OpenStack Japan Ops Workshop 2015/12 でのOpenStackを運用している各社の運用管理について

0) GMOインターネットの OpenStack public cloudサービスについて

We are offering multiple public cloud services.

Public Clouds

ConoHa by GMO

Z.com Cloud

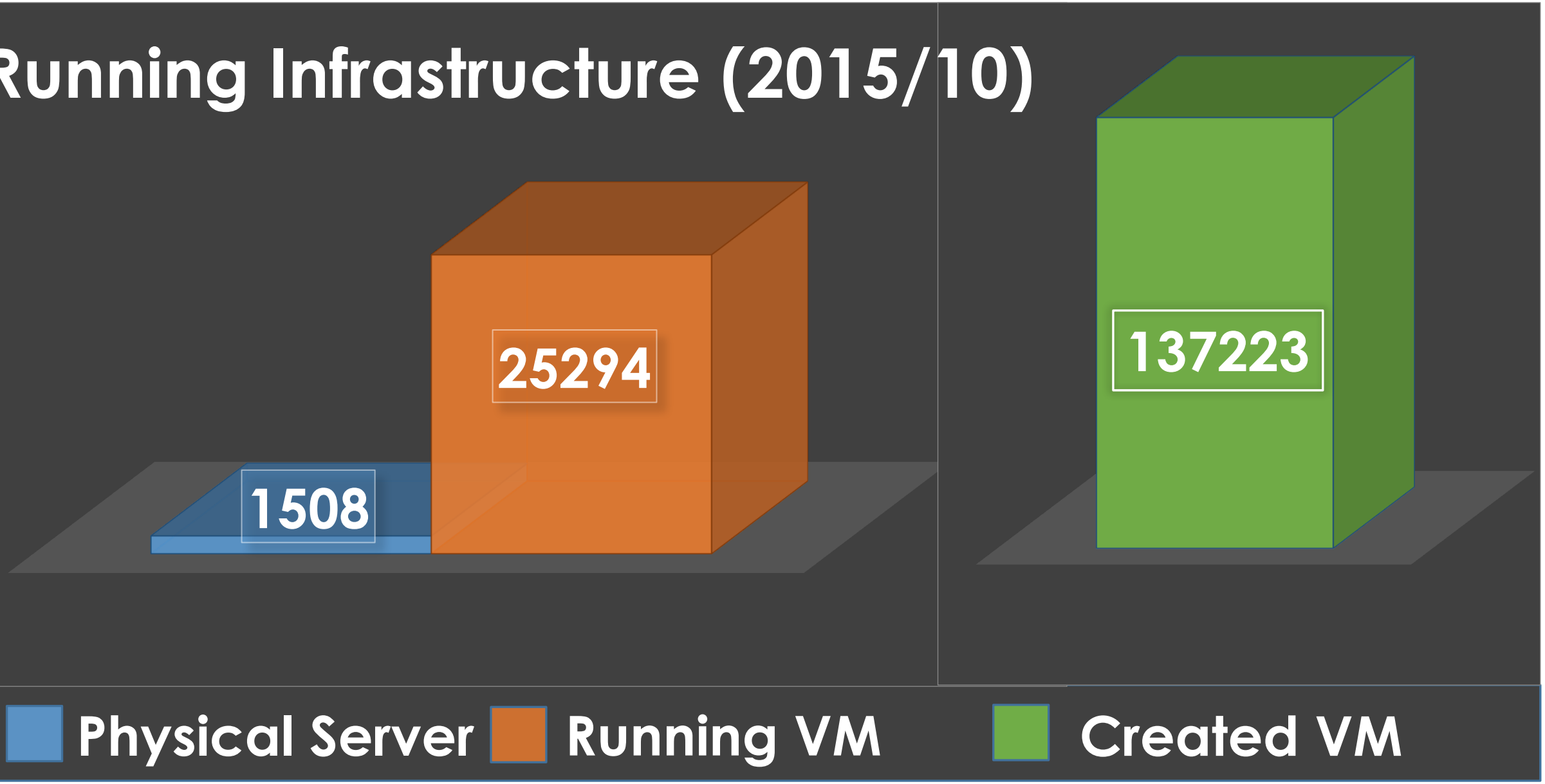
GMO AppsCloud

お名前.com VPS

お名前.com .com .jp .co.jp クラウド by GMO

openstack™ CLOUD SOFTWARE

Running Infrastructure (2015/10)



OpenStack service development team

Cloud service development team: Now(2016)

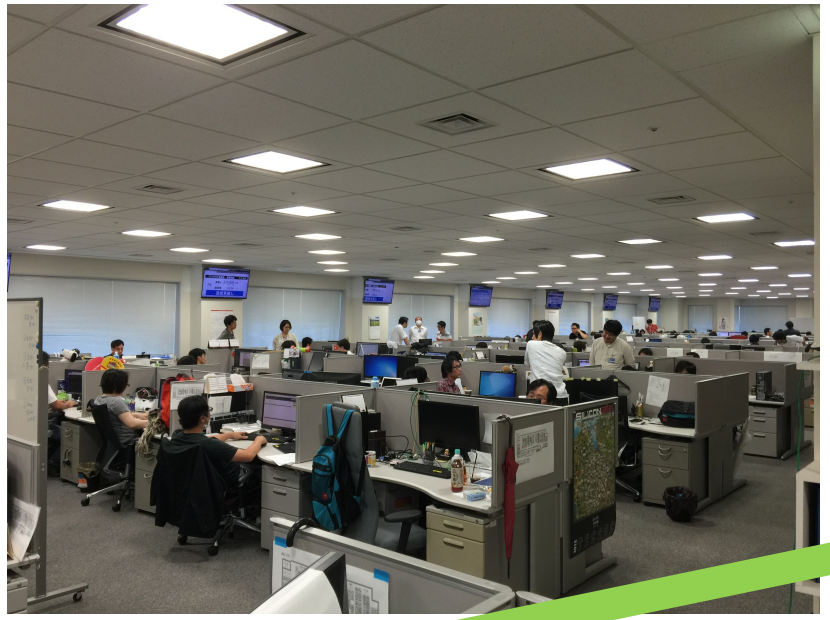
Cloud service development team: (abount 30 people)

- OpenStack Neutron team: 4 people
  - Neutron driver / modification / engineering
- Cloud API development team: 5 people
  - Public API validation program
  - OpenStack modification / scaduler programing / keystone
- Cloud Infra. development team: 11 people
  - Security engineering / glance driver / cinder driver / nova additional extensions / construction of OpenStack infra.
- Applicatoin cloud service development team: 5 people
  - Billing engineering / staff tools / GMO AppsCloud web GUI

Additional engineering team: many people (30 ~)

- QA Team / Server Engineering Team / GUI development Team
- Network Engineering Team / SaaS development Team
- CRM backend and billing Team

Cloud service development team: Office(2016) #1



Neutron Team
And
Cloud API Team

Cloud Infra. Team
And
AppsCloud Team

Cloud service development team: Office(2016) #2

Limied number of people.
But, we have to run a lot of OpenStack service clusters.

Service developmemt history by OpenStack

GMO Internet, Inc.: VPS and Cloud services

Onamae.com VPS (2012/03) :

http://www.onamae-server.com/

Forcus: global IPs; provided by simple "nova-network"

お名前.com レンタルサーバー by GMO

OpenStack Diablo
on CentOS 6.x

Nova network
Nova
Keystone
Glance

tenten VPS (2012/12)

http://www.tenten.vn/

Share of OSS by Group companies in Vietnam

Shared codes → tenten VN by GMO

ConoHa VPS (2013/07) :

http://www.conoha.jp/

Forcus: Quantam(Neutron) overlay tenant network

ConoHa by GMO

OpenStack Glizzly
on Ubuntu 12.04

Quantam
ovs + gre tunnel overlay
Nova
Keystone
Glance

GMO AppsCloud (2014/04) : http://cloud.gmo.jp/

OpenStack Havana based 1st region

Enterprise grade IaaS with block storage, object storage, LBaaS and baremetal compute was provided

GMOAppsCloud

OpenStack Havana
on CentOS 6.x

Keystone
Nova
Baremetal compute
Glance
Ceilometer
Cinder
Neutron
LBaaS

Shared codes → お名前.com クラウド by GMO

Onamae.com Cloud (2014/11)

http://www.onamae-cloud.com/

Forcus: Low price VM instances, baremetal compute and object storage

Glance
Keystone
Neutron
Ceilometer
Nova
Baremetal compute

ConoHa Cloud (2015/05/18) http://www.conoha.jp/

Forcus: ML2 vxlan overlay, LBaaS, block storage, DNSaaS(Designate) and original services by keystone auth

ConoHa by GMO + Z.com Cloud

OpenStack Juno
on CentOS 7.x

Designate
GSLB
Keystone
Nova
Glance
Ceilometer
Cinder
Neutron
LBaaS

GMO AppsCloud (2015/09/27) : http://cloud.gmo.jp/

2nd region by OpenStack Juno based

Enterprise grade IaaS with High IOPS Ironic Compute and Neutron LBaaS

GMOAppsCloud

Keystone
Glance
Nova
Ironic
Ceilometer
Cinder
Neutron
LBaaS

Shared cluster

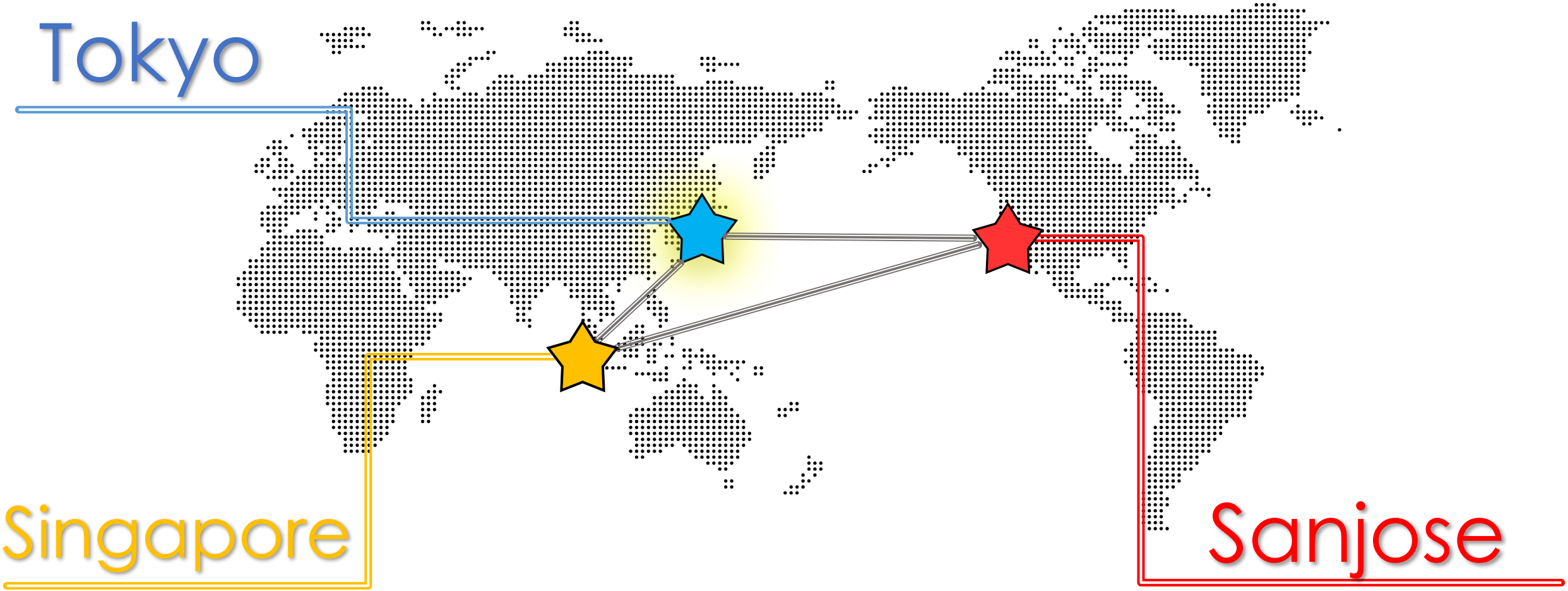
Swift cluster

Swift
Swift
Upgrade Juno
Swift
Swift
Swift

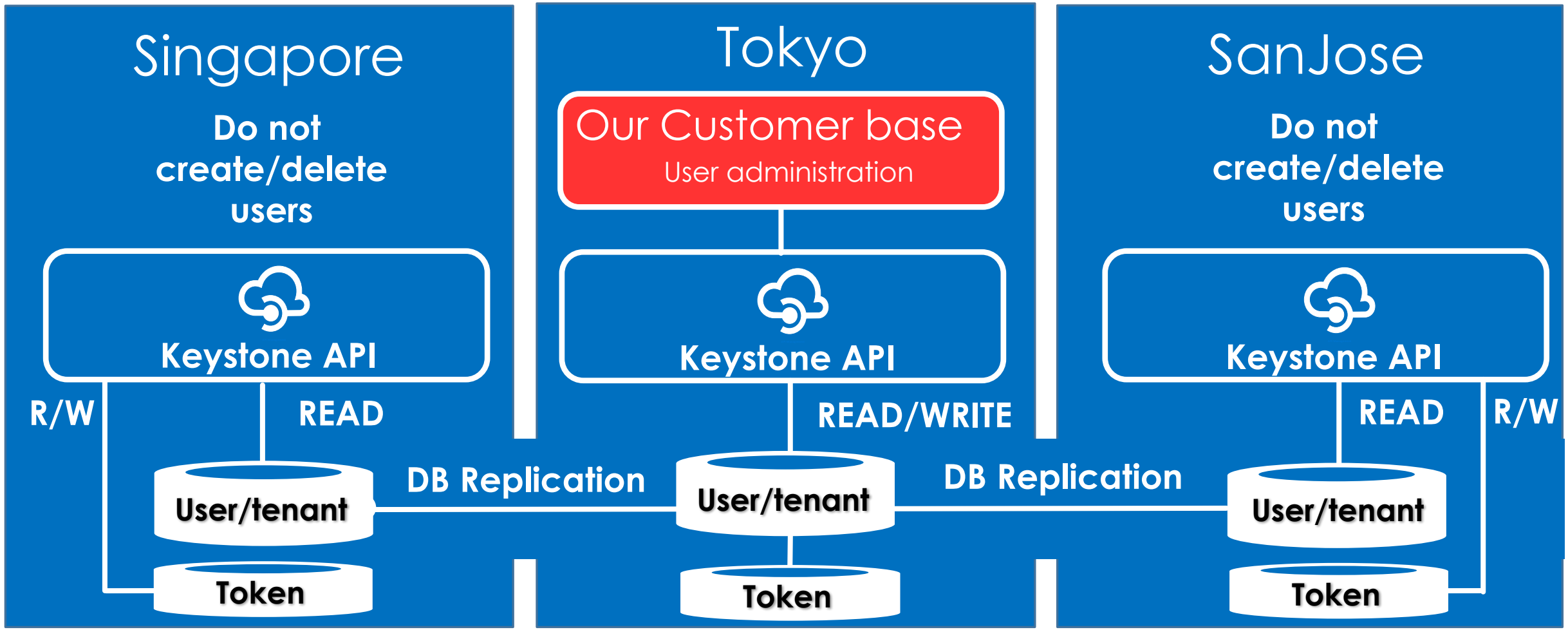
OpenStack Juno cluster:

- ConoHa (Juno) and Z.com cloud
- AppsCloud (Juno)

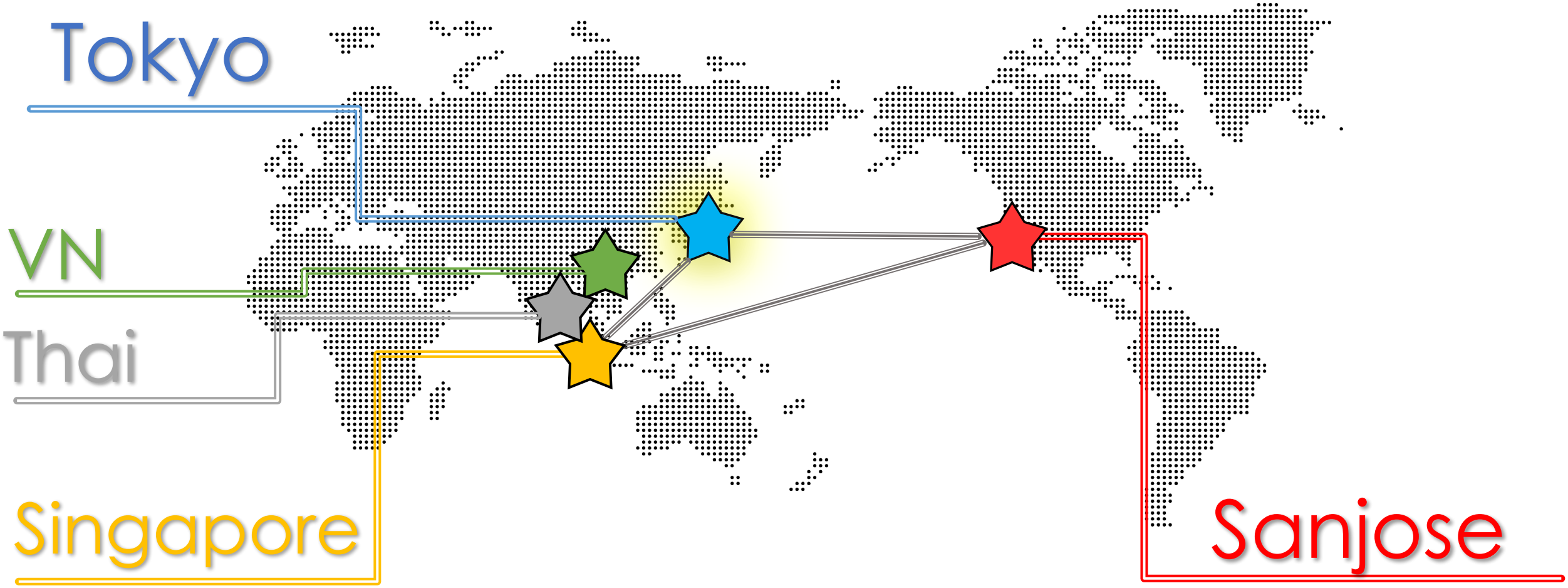
ConoHa has data centers in 3 Locations



User-registration is possible in Japan only



Now(2017); data centers in 5 Locations



OpenStack: we have many cluster

JUNO THE TENTH OPENSTACK RELEASE ConoHa by GMO

Mikumo ConoHa

Mikumo = 美雲 =
Beautiful cloud

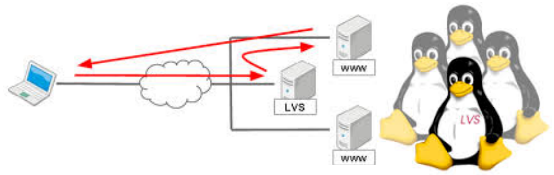
JUNO THE TENTH OPENSTACK RELEASE GMO AppsCloud

Mikumo Anzu

OpenStack Juno: 2 service cluster, released

JUNO THE TENTH OPENSTACK RELEASE ConoHa by GMO

- Service model: Public cloud by KVM
- Network: 10Gbps wired(10GBase SFP+)
- Network model:
  - Flat-VLAN + Neutron ML2 ovs-VXLAN overlay + ML2 LinuxBridge(SaaS only)
  - IPv6/IPv4 dualstack
- LBaaS: LVS-DSR(original)
- Public API
  - Provided the public API (v2 Domain)
- Compute node: ALL SSD for booting OS
  - Without Cinder boot
- Glance: provided
- Cinder: SSD NexentaStore zfs (SDS)
- Swift (shared Juno cluster)
- Cobbler deply on under-cloud
  - Ansible configuration
- SaaS original service with keystone auth
  - Email, web, CPanel and WordPress



nexenta Enterprise class storage for everyone

TOSHIBA Leading Innovation

JUNO THE TENTH OPENSTACK RELEASE GMO AppsCloud

- Service model: Public cloud by KVM
- Network: 10Gbps wired(10GBase SFP+)
- Network model:
  - L4-LB-Nat + Neutron ML2 LinuxBridge VLAN
  - IPv4 only
- LBaaS: Brocade ADX L4-NAT-LB(original)
- Public API
  - Provided the public API
- Compute node: Flash cached or SSD
- Glance: provided (NetApp offload)
- Cinder: NetApp storage
- Swift (shared Juno cluster)
- Ironic on under-cloud
  - Compute server deploy with Ansible config
- Ironic baremetal compute
  - Nexsus Cisco for Tagged VLAN module
  - ioMemory configuration

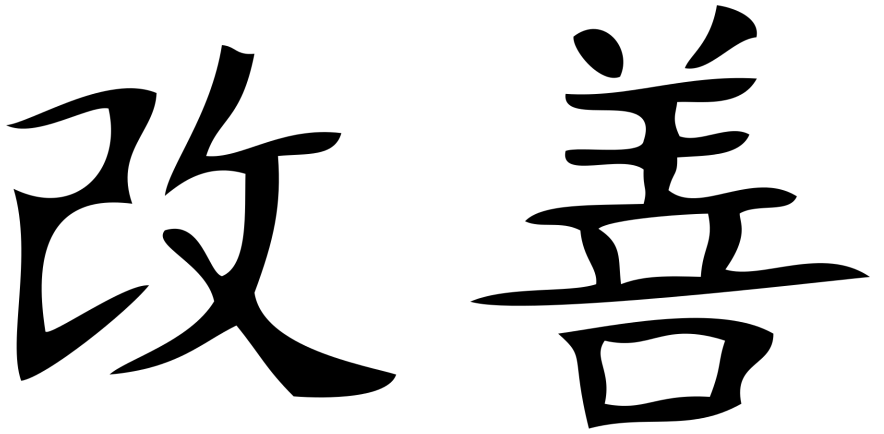
BROCADE

NetApp

Dark age for the Cloud suppliers

1）運用管理とはなんぞや

1)(システム)運用管理 とは?



- システム運用に関する管理
  - ➔ ITIL (Information Technology Infrastructure Library) などの例
    - ほぼデファクトのツール
    - ITサービスを管理するために、
      ITサービスの品質向上を目指す、
      中長期的なコスト削減を目指す(人的、金銭的)
  - いわゆる “PDCA” サイクル的なものを、回していく必要がある

システム運用管理 と カイゼン

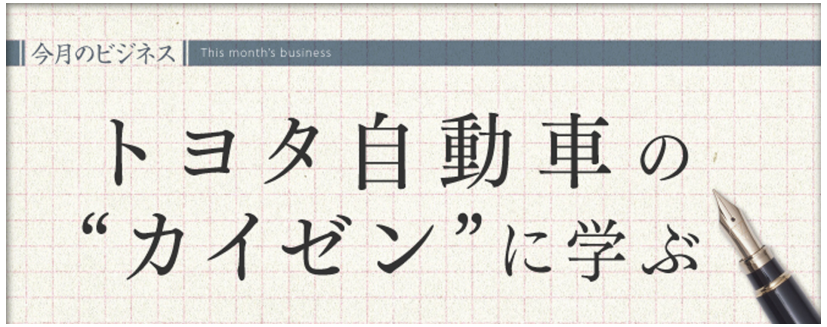
Implementation of Kaizen Result

BEFORE KAIZEN

AFTER KAIZEN

システム運用管理 と PDCA

「えっ、なんで?」



ACT

改　善

・経営層による見直し
・品質マネジメント
・システムの継続的改善

PLAN

計　画

・目的及び目標
・品質マネジメント
・プログラム　など

CHECK

点検及び是正処置

・監視及び測定
・品質マネジメント
・システム監査　など

DO

実施及び運用

・体制及び責任
・訓練、自覚及び能力
・運用管理　など

Ex）とある障害発生

お客のVPSのwebサーバのコンテンツが見えないと連絡

On call … …

お客のVPSのネットワークにDDoS攻撃来ていた!!

IPを上位ネットワークで止めてもらう

On call終了

ふぅっ

Ex）とある障害発生

お客のVPSのwebサーバのコンテンツが見えないと連絡

On call … …

お客のVPSのネットワークにDDoS攻撃来ていた!!

IPを上位ネットワークで止めてもらう

On call終了

ふぅっ

ちょっ、まぁっ、ここで終わったら、次にDDoS来た時にどうするんだ!!

Ex) とある障害発生 + 記録 + 改善策 + 記録

お客のVPSのwebサーバのコンテンツが見えないと連絡

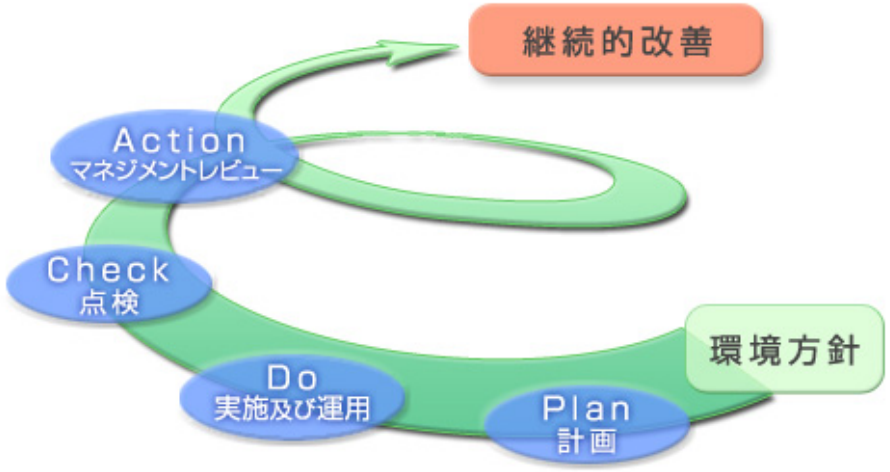
On call … …

お客のVPSのネットワークにDDoS攻撃来ていた!!

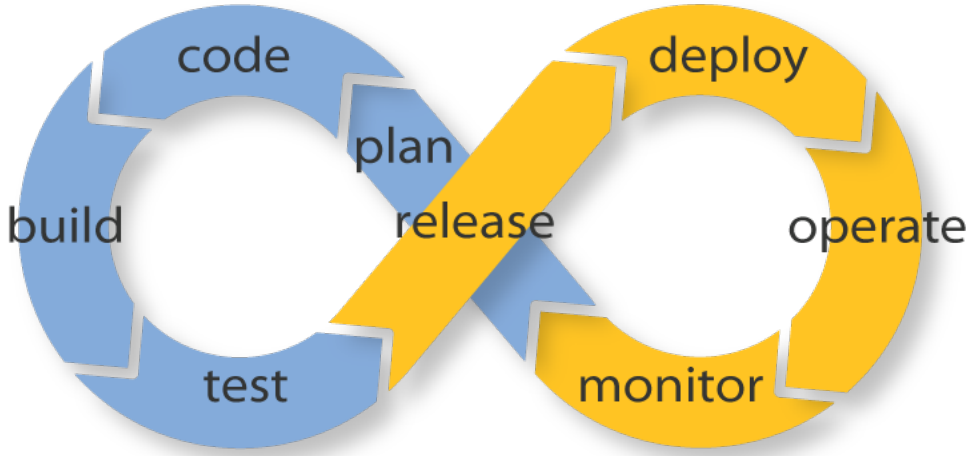
IPを上位ネットワークで止めてもらう
➔対応内容を記載
On call終了
➔対策を調査 ➔ http://blog.sflow.com/2013/03/ddos.html
VPSのovs sflow をElasticSearchに入れて、AlertAction設定
AlertActionで上位ルータにdev nullに3分間落とす
対策内容を共有、類似のトラブルですぐに確認できるように

PDCA: (Plan, Do, Check, Action(Adjust))

- 継続的改善



DevOpsへの道



Endless Possibilities: DevOps can create an infinite loop of release and feedback for all your code and deployment targets.

システム運用管理ソフトウェア とは?

- システム運用に関する管理 ➔ カイゼンを回して効率化などを得る

この場では、事例、方法論などの知見を得ることで、何らかの改善ができればよいと思います

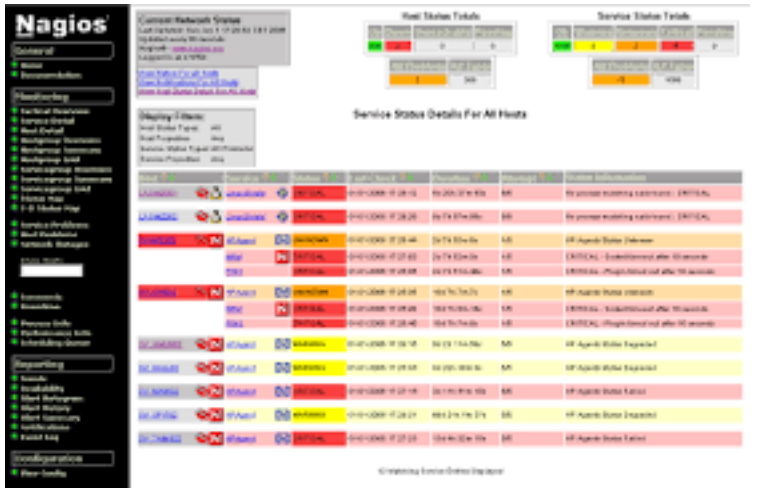
以下の内容(要素)を含みます ➔ ネタはたくさんあります

- Monitoring System, Security System
- Notification(ChatOps, PushNotify), redmine(Tiket), RTFM
- CMDB, IPAM, SDN, OpenStack, CloudStack
- Automation Tools, Infra-spec Tools, Jenkins(Job)
- Documentation Tools (Wiki, RTFM, Jupyter)
- … …

2）これまで運用管理に使っていたツールについて

2) これまでの運用管理に使っていたツール

- 1) 監視ツール: Nagios
  - Version: 2.0b6(Solaris), 3.0(Solaris), 3.5.1(Nagios Core)
  - OS環境が様々
  - Solaris 9, 10, OpenSolaris
  - Linux: CentOS 5.x, CentOS 6.x, CentOS 7.x, Ubuntu 14.04
  - Notification:
    - Mail:
    - IRC: #nagios channel へのircbot (
    - Nagstamon: (Windows): https://nagstamon.ifw-dresden.de) https://github.com/HenriWahl/Nagstamon
  - Graph: nagiosgraph (rrd files)



入社したときに初めてさわったNagios(ver. 2.0b6)が現役監視　➔ Solarisのシステム

2) なぜNagios 2.0b6とか残っているの?

なぜ?

- Nagios自体がほとんど進化していない?
- Nagiosの残念な開発体制と、たくさんのForkの発生
  - Nagios自体、仕様がほとんど変わらないので、
    稼働が始まればupgradeする必要性を運用的に感じなかった
- APIなどの不統一、オレオレFork
- 監視対象もSolarisでgccでコンパイルが通れば問題ない
- 監視数が増えないので、スケールアウトとか必要なかった
- グラフもrrdデータなので

なので、問題なかった(いままでは)

Nagios plugin(監視)はシンプル

Nrpe agentプログラムを使う場合、監視スクリプトは以下のステータスコードとテキスト標準出力を返せばよい

Nagios Exit Codes

Exit code	status
0	OK
1	WARNING
2	CRITICAL
3	UNKNOWN

Nrpe agent ➔ Nagiosクロールへの通信も、同様な処理

監視が作りやすい

2-2) これまでの運用管理に使っていたツール

- 2) Hardware監視ツール:
  - Director (IBM, Lenovo)
  - SIM [System Insight Manager] (HPE)
  - OpenManage (Dell)

導入されたのは、お名前VPSサービス開始後(初めてLinux採用)

>> IBM IA Serverが初めて導入された

製品のDefaultでIMMというIPMIの拡張管理ポートがあった

IPMI関連ツールやコンソールの有用性から、ここから広まった

Hardware監視ツールが無料で利用できたというメリットも

2-2) HW監視ツール

- IBM Director
  - DB(IBM DB2 or MS SQLなど)の負荷などで、構成の工夫は入ります
  - Agent less監視 <<= ここ重要
- SIM (HP)
  - PostgreSQL
  - Agent必要 (HP ProLiant Gen9 あたりで、できるようになった?)
  - HPは買わなくなりましたw
- Dell
  - MS SQL Server
  - Agent less監視 <<= ここ重要

2-2) Agentがあると、なにが問題か?

- SIM (HP)
  - HPは買わなくなりましたw ← なぜ?
- Agentが強制再起動を発動 (ASR)
  - Hung upを検知した時に、強制再起動
  - 外部の温度で、強制再起動
  - 思わぬDefault設定 www
  - ➔ 結局Agentを停止することに
- その他、SmartArrayトラブル等、みんな運用疲れ

2) これまでの運用管理に使っていたツール

- 3) RTFM: Request Tracker
  - 入社当初のドキュメント/チケット管理ツール(2007, 2008)
  - Read the Fucking Manual
  - Site: https://bestpractical.com/rt-and-rtir
  - Document: https://docs.bestpractical.com/rt/4.4.0/index.html
  - Apache + mod_perl (環境はもちろんSolaris)
  - ドキュメント管理とチケット管理、Task Flowのツール

程なくして、redmineのチケット管理が入ったので、しばらくドキュメント共有ツールとしてつかつていました

2）これまでの運用管理に使っていたツール

- 4) redmine:
  - チケット、Task Flow管理
    - そもさん(お問い合わせ管理ツール)
    - SQIP2 (障害管理ツール)
    - Liveup/メンテナンス管理
    - チーム間業務依頼管理
    - Abuse管理
    - QAバグチケット管理
    - プロジェクト管理システム(工数管理)

APIでチケットを投入したり、参照したりでいろいろ他のシステム連携

入社してからIDC OPEチームが導入

現在のシステム本部と事業部のTask Flow基盤として

2) これまでの運用管理に使っていたツール

- 5) dokuwiki:
  - プロジェクトのドキュメント、作業メモ、仕様書記載
  - RTのテキストのみから、画像、添付書類がつかえるように
  - XML-RPC (JSONは標準では無い)

入社してからIDC OPEチームが導入

RTからはドキュメント・データ移行

2) これまでの運用管理に使っていたツール

- *) その他:

- 設備機器管理ツール(開発部作成 .Net)
  - 開発部で作成: ラックの管理なども含む

- オンコール管理ツール(開発部作成 .Net)
  - 電話/メールの連絡先と、ローテーションスケジュールの管理

- IRC
  - ChatOpsの原点、作業時の連絡など
  - NagiosからAlert Notification

3) “ConoHa”をマルチロケーションで運用するにあたって、再検討した最新の運用管理ツールについて

3) ConoHaをマルチロケーションで運用時の再検討

- A) 監視ツールのNagiosの数がサービスごとに発生
  - NotifyにMailを使っているので、莫大なWARN, ALERTメールが発生
    - MialをNotifyにすると、通常業務のMailに負荷がかかり、Mailの障害という二重障害が発生
  - ロケーションが違うと、メールの到達性に疑問

- B) マルチロケーションをどう管理しやすくするのか
  - 画面上統一管理できるか

- C) 遠隔地は自社で運用するとは限らなくなる(日本だけじゃない、現地法人)
  - グループ会社、協力会社が使っても良いようにユーザ管理できる
  - Html5 SSHなど、web画面で作業が終端できる、証跡 (記録)

3) 再検討ツールたち

- 1) Nagios fork系、インスパイア系
  - Nagios監視プラグインが流用できる

- 2) 新たな構成
  - Hatohol + zabbix / Monascaつかえる?
  - Hatoholにより、統合できそう
  - Gateone pluginとか作れるかも(zabbixからclick to html5 ssh)
    - http://tech-sketch.github.io/hyclops/jp/
      TISが作ったFork : HyClops for Zabbix

- 3) そのままNagiosを個別に立てる
  - 方法論は同じだが、「カイゼン」全く無し <<= 抵抗勢力はこれが良いとw

3) 再検討: A) Nagiosインスパイア系: Ichinga

- Ichinga
  - https://www.icinga.org/
  - CLIも強力、分散監視できる
  - Nagios pluginをそのままつかえる (pluginが多い)
  - Webと本体がわかれている、Timelineてきなインターフェースもよさ気
  - 有償サポートあり、HPE Helion OpenStackもIchingaを採用
  - WebはPHP？

- 以外にインストール面倒
- 監視設定処理の概念がNagiosを引き継いでいるので、以外に面倒
- DBを使う

3) 再検討: A) Nagiosインスパイア系: Sensu

- Sensu
  - https://github.com/sensu/sensu/
  - Ruby、RabbitMQ, Redis, JSON
  - メッセージ志向 (スケールアウトできるように)
  - メトリックス送信先で、グラフ書いたりする(Graphite, Librato, etc)
    - グラフ機能は内蔵されていない
  - Chef, puppet, ansibleでのWorkFlowで監視を投入できる
  - GUIは “uchiwa.io” : https://uchiwa.io/
  - HandlersでNotifyアクション, Graphiteなどへのメトリクス POST(グラフ更新)

- 素晴らしいんだけど、学習コストいまは高そう(ConoHaリニューアル時)、Yahoo Japanは使っている

3) 再検討: A) Nagiosインスパイア系: Shinken

- Shinken
  - http://www.shinken-monitoring.org/　　shinken.io
  - https://github.com/naparuba/shinken/
  - PythonでNagiosをよりEnterpriseに作りなおした感じ
  - マルチデータセンタに適合しやすい構成
  - Graphite, Nagvis, PNG4Nagios, Nagios OLD cgi webにも!! 対応
  - Livestatus based API対応

- あぁ、普通にNagiosの新しいバージョンとしては良いかも
- Nagios過ぎて、若干引く感じ

3) 再検討: Hatohol + zabbix, Nagios

- Hatohol
  - http://www.hatohol.org/
  - https://github.com/project-hatohol/hatohol
  - Miracle Linuxが主要メンバーで開発しているOpensource
    - なので、githubも日本語のissueがある
  - Zabbix or NagiosのFrontendとなる
  - Metricを実際にするzabbix, Nagiosをまとめるであり、Notify、Actionを統合できるツール

- OpenStack連携の話をイベントの時に聞いた

➔ あ、よさそう

4）Hatohol ＋ zabbix 構成を選択した理由

4) Hatohol + zabbixを選択した理由

- A) 監視ツールのNagiosの数がサービスごとに発生
  - Notifyの場所をHatohol一箇所にすることで、Action(ex: チケットを作ったり、chatに通知したり、メールに投げたり)を作りやすくする
  - NagiosもZabbixもCeilometerもメトリックとして一元化できる、だろう

- B) マルチロケーションをどう管理しやすくするのか
  - Hatoholに情報ポータルを一元化でき、zabbixのスケールアウトとして活用できる (zabbix APIで接続) (取得結果はHatohol内部のDBにcache)

- C) 遠隔地は自社で運用するとは限らなくなる(日本だけじゃない)
  - HAPI(Hatohol Arm Plugin Interface)経由で認証を作ればできる、らしい
    - グループ企業は、web auth APIに対応すれば良い

zabbix

- エージェントレスでssh baseでもいけますよ(by 工藤) (zabbix ver. 2.2系当時)
- GUIはモダンでもないけど、APIが公式である (zabbix API)
  - ライブラリ結構ある: https://www.zabbix.org/wiki/Docs/api/libraries
  - Ex) Php: PhpZabbixApi : http://github.com/confirm/PhpZabbixApi
- Nagiosみたいに分裂、発散せず、zabbix.org としてまとまっているのが良い
  - 情報も探しやすい (オープンソース版、商用版) 、情報が多い重要
  - オープンソース活用において、商用とコミュニティが活発である (重要)
  - Forkしたくはないけど、pluginぐらいなら作るのはOK
- まあ、そろそろ他の監視ツールも使ってみたい

Hatohol + zabbix, nagios

- Hatoholにより、監視ツールは異なっても、おなじようにいけるだろう
- Nagiosのところは、Sensu(ruby), Shinken(python), Ichingaになっても、なんとかなるかもとかんがえられる (置換可能)
- Nagiosのところは、**”live status api”**がよいかな(2015/05 導入当時)

MK Livestatus. Get Nagios data with Python API.

This entry was posted in check_mk | MK Livestatus | plugin *and tagged* check_mk | LQL | mk livestatus | plugins | python | unixcat *on 28/02/2015 by distractedman1*

Introduction.

MK Livestatus provides a standard API for accessing Nagios data in various programming languages: Python, Perl and C ++.
The API modules and sample documentation is available for use in these languages and is enough to start testing programs. We must have prior knowledge of Accessing Nagios data with “unixcat” and LQL.

```
import livestatus
import sys

cmk_livestatus_nagios_server = "localhost"
cmk_livestatus_tcp_port = 6557
host_to_find = "srv0001a"

try:
    socket_path = "tcp:%s:%s" % (cmk_livest
except:
    sys.exit(1)
```

The route to the documentation and the source / libraries normally is in a OMD setup in /omd/versions/default/share/doc/check_mk/livestatus/api/. If we do a direct install of check_mk or MK livestatus standalone the default path usually is the following: /usr/share/doc/check_mk/livestatus/api/.

Hatohol + nagios livestatus API

月日は流れ、秋ごろ

Hatohol + zabbix or nagios

- (2015/Sep) Issueが上がる

- https://github.com/project-hatohol/hatohol/issues/1549
- Mnakagawa-san作る

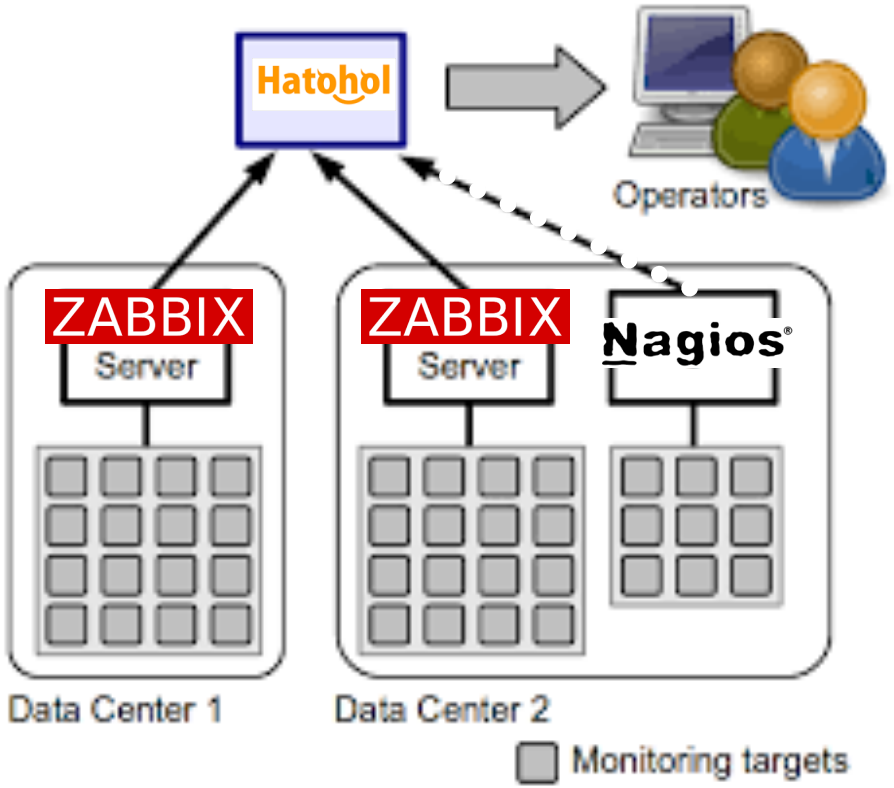
- 実装された:
  https://github.com/project-hatohol/hatohol/blob/master/server/README.md
  - これにより、NagiosをNDOUtilsにマイグレしなくても、Nagios Livestatus APIの設定のみで連携できるように

    (我々は、zabbix導入後ですが、OKだなと)

5) Hatohol + zabbix構成

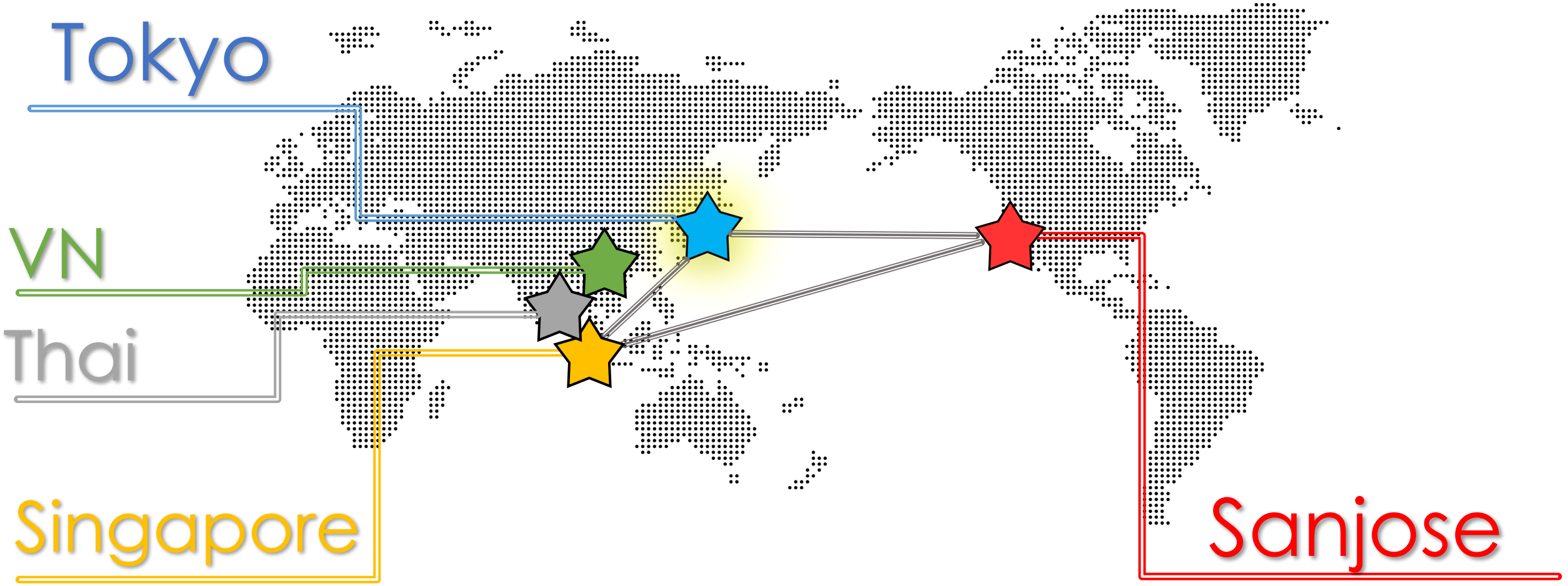
5) Hatohol + zabbix構成 (2017/02/17現在)

- 各リージョンのOpenStack cloud clusterの監視 : zabbix
  - JP (Tokyo)
  - US (San Jose)
  - Singapore
  - Vietnam (Hanoi)
  - Thailand (Bangkok)



- 情報の集約 : Hatohol (JP (Tokyo))
  - Notification Action : redmine
  - 1 AlertごとにHatoholがredmineにアラートチケットを発行(Action)
- 証跡 : redmine
  - アラートをチケット =>> メールとしてNotify

Now(2017); data centers in 5 Locations



5) zabbix

- zabbix単体でのalert通知は利用しない
  - Zabbixはバックエンドが Databaseであり、alert履歴など貯めないことでなるべく軽くすることを考えておく
  - Zabbixの古い履歴は、Databaseが肥大化する
    - DB (MySQL, PostgreSQL など)が膨れるので、履歴データを定期的に削除
      - 履歴はredmineに「証跡」として残すので、気軽に消せる
    - グラフも紐付いているので、グラフを残したい場合には、別途グラフツールを考える
    - Zabbix ➔ Graphite + Graphana など
  - OpenStack clusterのリージョン(ロケーション)ごとに設置する

5) Hatohol

- Zabbixからの情報の集約とActionの生成に専念させる
- Action :
  - 現在は 「Alert発生アクション」をredmine チケットに
  - 「Alert収束(終了)アクション」を通知してほしいとのユーザリクエスト
    - Nagiosにはある機能 (今後の課題)
- 今後の活用
  - Hatohol APIとchatops (chatworks, slack)との組み合わせ
- オープンソースであるので、開発コミュニティに参加して機能改善していく
  - 運用監視の改善プロジェクト

www.hatohol.org

Hatohol | Hatoholの概要 | スクリーンショット | ドキュメント | プロジェクトへの協力

Hatohol

Hatoholとは

Hatoholは、システム監視やジョブ管理やインシデント管理、ログ管理など、様々な運用管理ツールのハブとなるツールです。
現段階ではシステム監視ツールの統合機能を持っており、複数のZabbixとNagiosで監視を行っている場合に、それらの蓄積する監視情報を集約して表示する機能を持っています。詳細はこちら

更新情報・お知らせ

2017/01/13	Hatohol version 16.12が本日リリースされました。(インストールドキュメント)
2016/5/13	Hatohol version 16.04が本日リリースされました。(インストールドキュメント)
2016/1/31	Hatohol version 16.01が本日リリースされました。(インストールドキュメント)

Japan's Leading All-in Provider of Internet Services GMO

GitHub, Inc. [US] https://github.com/project-hatohol/hatohol

This repository Search Pull requests Issues Gist

project-hatohol / hatohol

Unwatch 39 Unstar 86 Fork 35

Code Issues 127 Pull requests 1 Projects 0 Wiki Pulse Graphs

A unified manager of monitoring software http://www.hatohol.org/

17,476 commits **75** branches **31** releases **21** contributors

Branch: master New pull request Create new file Upload files Find file Clone or download

masa0612 committed on **GitHub** Merge pull request #2510 from project-hatohol/new-overview-image2 ... Latest commit 3c74a7a 3 days ago

client	Merge pull request #2491 from project-hatohol/work-for-splitting-#2370-...	a month ago
data	Update init scripts for CentOS and systemd.	2 months ago
debian	license: Change hatohol license from GPLv2 to LGPLv3.	2 years ago
doc	[Doc] Renew an overview image of Hatohol	5 days ago
server	replace UsedCountable of AMQPConnection by shared_ptr	11 days ago
setup-with-ansible	[ansible][centos7] Use epel-release package in base	9 months ago
test	[test][client] Execute gulp more later test phase	a year ago
wercker	[Wercker] Try to use CentOS 6.7 DockerHub image	a year ago
.dir-locals.el	[Emacs] Add .js configuration	3 years ago
.gitignore	[.gitignore] Update files to be ignored.	2 months ago
.jshintrc	[lint][.jslintrc] Use English for comments	a year ago

TOP

お申し込み・お支払い

コントロールパネル基本操作

オブジェクトストレージ

テンプレートの使い方

KUSANAGI

Drupal

Hatohol(16.04) の使い方

※このドキュメントはミラクル・リナックス株式会社から提供を受けたものです。

※Hatohol(16.01)をご利用の方は こちら をご確認ください。

Hatoholは複数のZabbixやNagiosといった監視ツールの情報を統合するオープンソースソフトウェアです。

Hatohol16.04のアップデートでは、イベントの取得方法が代わりよりセキュアに各ツールと連携が可能になったほか、イベントにコメントを書き込む機能などが追加されています。


Fin.
ConoHa
by GMO

Appendix: OpenStackの運用管理

Appendix

6) (参考) OpenStack Japan Ops Workshop 2015/12

- 2015年はじめて東京でOpenStack Summitが開催(10月)
- Ops Workshop自体はSummitの中でもこれまでは開催されていたが、セッションが同時開催だと参加しづらいという側面も
  - Mid-Cycleとして Summitとの中間時期にも開催されることに
- OpenStackの開発へのFeedback(blueprintを書けないopsの意見取り込み)や、ドキュメント開発(ドキュメント作成は開発と同等となった)への取り組みへの反映
- OpenStack Summitとは日本ではずらして開催
  - 東京(7月)、沖縄(12月)
  - 次回開催は OpenStack Days Tokyo in July 2016

Appendix

OpenStack ops docs（ドキュメント）

- Ops Guide
- HA Guide
- Security Guide
- New Architecture Design Guide
- Networking Guide

http://docs.openstack.org/ops/

電子書籍データ(無料)

http://docs.openstack.org/ja/openstack-ops/content/

html閲覧 オペレーションガイド(ja)

http://docs/openstack.org/sec/

html閲覧 セキュリティガイド(en)

Appendix

OpenStack ops 日本でどんなツールを使って運用しているか

http://superuser.openstack.org/articles/okinawa-openstack-ops-workshop-rides-wave-of-interest

Okinawa OpenStack Ops Workshop rides wave of interest

Etherpad (議事録まとめ)

https://etherpad.openstack.org/p/JP-Ops-workshop

Appendix

ELKつかってますか? (日本のユーザ向け)

Noooooooo.

➔ なぜか?

日本では Logstash よりも Fluentd が好まれます

- つまりELK(E: ElasticSearch, L: Logstash, K: Kibana)ではなく、EFK(E: ElasticSearch, F: Fluentd, K: Kibana)である
- 最近は”Beats”というものもある(Goで書かれているのでちっさく、ツールが用途別になっている）>> EBK？

説明スライド

http://www.slideshare.net/td-nttcom/collect-summarize-and-notify-of-openstacks-log

Collect, summarize and notify of OpenStack's log

Appendix

Logstash vs. Fluentd

Logstash

- Javaである（メモリがっ…）“JRuby” runtime、スケールしない?
- Logのcollect + forward
- Plugin
  - Input/Output/codec/filterなどのpluginがある(ruby)

Fluentd

- ちっさい、軽い? Norikraなど中間処理に渡して再処理などgateway
- Plugin
  - Online processingがある ➔ Norikra（ここは”JRuby”なんだが）
  - Collectorでcounter組み込み(grep counter, datacalculator, etc.)

Appendix

opsツール(Workshopアンケートより)

Log monitoring tol

- Logstash: 0
- Fluentd: 4
- Beaver: 0 << https://github.com/josegonzalez/python-beaver
- Rsyslog: 6
- Splunk: 4
- ELK:

Splunkは有償ソフトだが、4社も使っているとのこと(使いやすい)

Appendix

log処理で何をやると便利か?

- 串刺しで検索して動きを見る: 共通IDの検索
  - Request ID
  - Tenant ID
  - User ID
  - その他 object(flavor, volume, image) ID

Appendix

OpenStackの監視系、アンケートより

- Kibana: 1
- Grafana: 2
- Collectd: 1
- Nagios: 1
- Zabbix: 9
  - << スケールしない(170,000 metrics)、グラフがpoor
- Sensu: 2
- Monasca: 0
- HRForcast: 1 (data visualizer,
  - GrowthForcastがRRDToolに依存しているが、
    HPForcastはCSV(TSV)とかでOK、

Appendix

OpenStackの監視系、アンケートより

Ceilometerはつかつている?: 4 企業 yes

- Backend: Mongodb
- なににつかっているの?
  - Billing: 1 (GMOのみ)
  - HeatでのAutoscaleのAlertingに使っている

VMの中身の監視は?

- Nagios, Sensuなどの通常監視
- Libvirtdからの外部からの監視
- Guest agent
  - Qemu-guest-agent (qemu経由でguestの情報取得)
  - https://github.com/rackerlabs/openstack-guest-agents-unix

Appendix

OpenStackの監視系、アンケートより

CMDB(構成管理DB)とか使っている?

- 開発CI, alerting, notification system
- 自社CMDB : 3 企業
- CMDBでインフラを管理(NW(IPAM), Server)
  - << dynamic Inventory?

その他使っているもの

- Spark
- ElasticSearch
- InfluxDB +1
- graphite

Appendix

OpenStackのlogのこれから

- “oslo.log” という共通ライブラリ化により、標準フォーマット化
- Traceback logは複数行にまたがっている << Parseしづらい
- Error codeのoslo.logでの統一化

Appendix

OpenStack deployment tips

- Puppet : 3
  - OS-puppet : 0
- Chef : 4
  - OS-Chef : 0
- Ansible : 7
  - OS-ansible : 1
- Juju : 2
- Fuel : 1

それぞれ、運用管理ツールを兼ねているものがある

Fuelなど

Appendix
その他

詳しくは、以下を見てください。箇条書きになっています

Etherpad (議事録まとめ)

https://etherpad.openstack.org/p/JP-Ops-workshop