Zabbix Non-Certified Introductory Training '22

2022年11月18日

NTTコミュニケーションズ株式会社

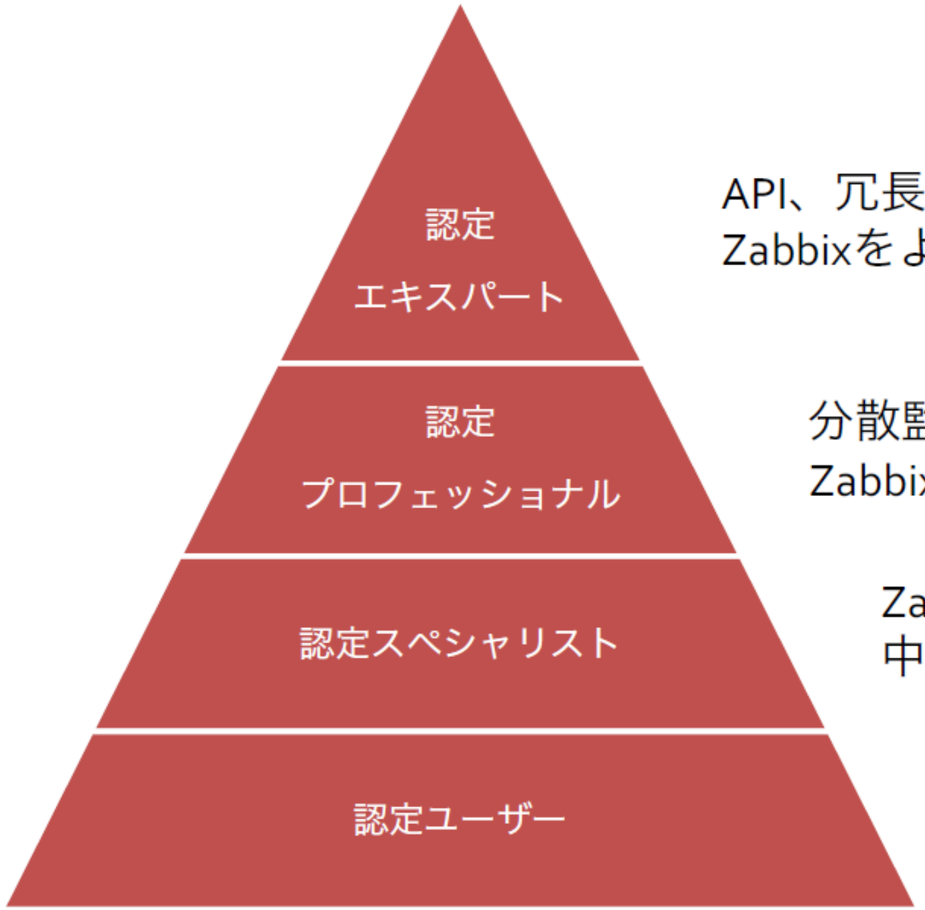
Zabbix Certified

User
Specialist
Professional
Expert

入門
API

Training

Zabbix 認定トレーニングコース



API、冗長化、DBパーティショニングなど
Zabbixをより活用するための機能

分散監視や大規模なシステムの監視
Zabbixの拡張的な機能

Zabbixのインストールと設定、基本機能
中小規模のシステムの監視

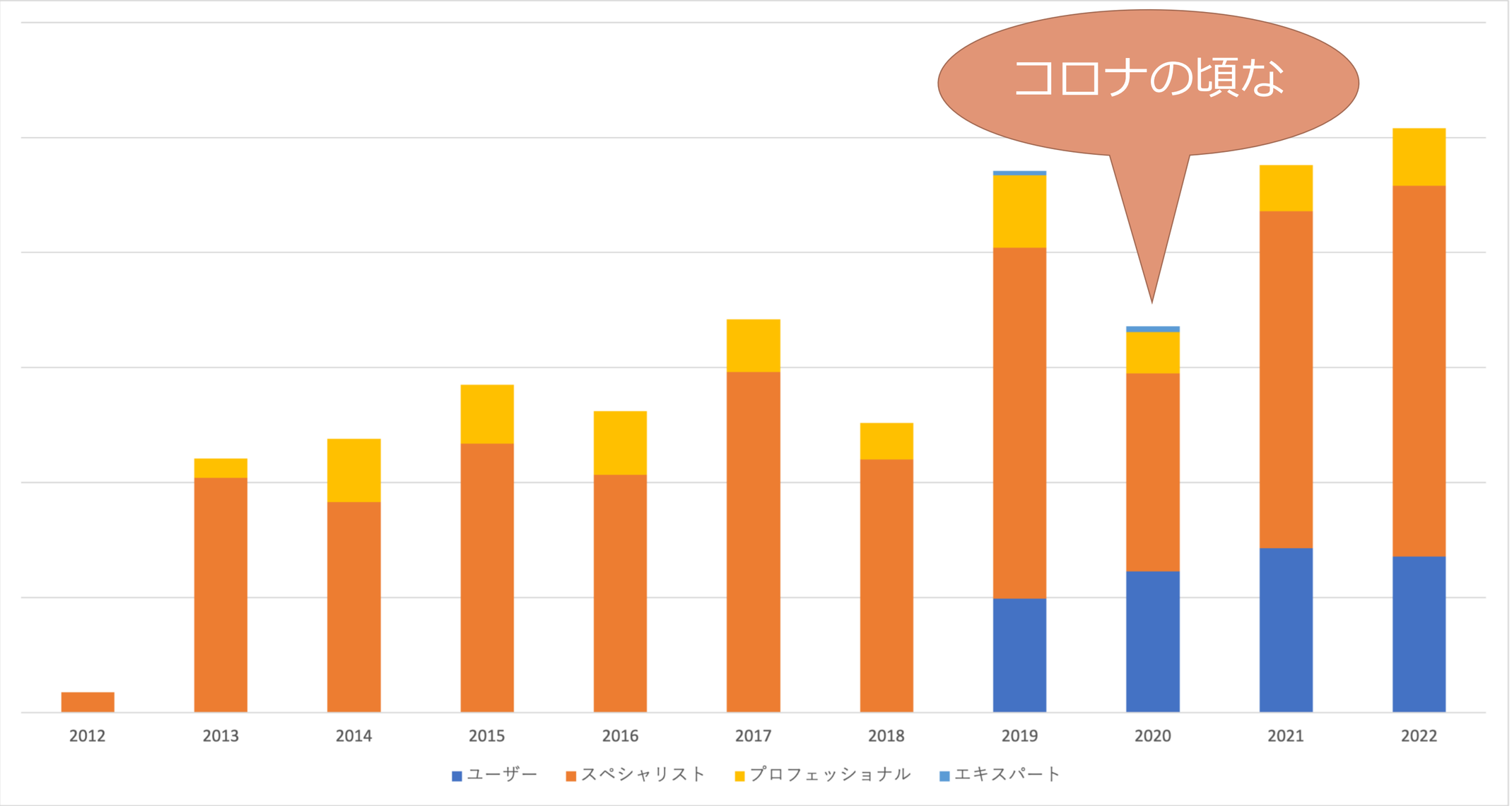
Zabbixの基礎的な知識と
Webインターフェースの利用

日本独自

入門 Zabbixを利用し始める方がZabbixの基本的な操作設定を学ぶことができるコース

API プログラミングを行いながら実際に動作するアプリケーションを作成し、
APIの動作の基礎から実践的な活用までを学ぶことができるコース

みなさまのおかげです



Zabbix認定トレーナーの紹介

トレーナー紹介【入門コース】 日本独自

Zabbix入門

Zabbixを利用しはじめる方が
Zabbixの基本的な操作設定を学ぶ
ことができるコース

3時間

要件
なし

18,000円（税抜）

申込み

詳細情報

渡邉 隼人
Zabbix Japan LLC

2011年から、約5年Zabbixを使用した運用及びR&Dを行ってきました。2016年4月より経歴を活かしZabbix Japanに入社し、サポートエンジニアとして従事しております。講師としては社内外でハンズオンやセミナーの登壇を行っております。

Zabbix 1.6から使用していますが、進化を続けるZabbix。その機能や性能の向上は導入頂いている皆様の評価やご要望があってこそだと思っています。これから初めて触ってみる方々、運用監視についてまだ足を踏み込みきれていない方々に魅力を伝えていきます。

トレーナー紹介【APIコース】 日本独自

Zabbix API

プログラミングを行いながら実際に動作するアプリケーションを作成し、APIの動作の基礎から実践的な活用までを学ぶことができるコース

2日間

要件

Zabbix認定スペシャリストコース修了証明書
または、Zabbix認定スペシャリストコース修了相当の知識

100,000円（税抜）

申込み

詳細情報

寺島 広大
CEO, Zabbix Japan LLC

現在10年以上のZabbixの経験があります。Zabbix LLCにてテクニカルサポートエンジニアとして働き、2012年10月からはZabbix日本支社のCEOとして支社の経営を行っています。2005年に設立したZabbixの日本コミュニティーの創設者でもあります。2010年には、Zabbixの日本語書籍の執筆も行っています。日本で最もZabbixの経験を持つエンジニアです。

トレーナー紹介【認定ユーザーコース】

Level 1

Zabbix認定ユーザー

Zabbixのフロントエンドの使い方、およびZabbixのポテンシャルをご紹介するコース

1日間

要件
なし

50,000円（税抜）

申込み

詳細情報

小野 博喜
NTT Communications

2007年からZabbixの立上げ初期から検証＆構築業務に従事しており、豊富な構築経験と運用管理のノウハウには定評がある。また、クラウド基盤の監視案件にも携わり、豊富な経験や技術力に基づく仕事には定評がある。温和でフレンドリーな性格なので、受講者の良き相談相手として相性抜群！

小野氏がZabbixを薦める理由
10年ほど前と比べるとZabbixは、オープンソース監視ツールの中でも、かなりメジャーなツールになっております。今後、更なる機能拡張や機能改善を見込まれており、ますますZabbixの利用現場が増えていきますので、是非、Zabbixをご利用してみてください。

トレーナー紹介【認定 SP、Pro、Ex】

Level 2

Zabbix認定スペシャリスト

Zabbixの主要機能、インスト
ル、設定、メンテナンス方法
得するコース

3日間

要件

オペレーティングシステムの
と上級コンピュータリテラシ
Zabbix認定ユーザーコース修
当の知識

150,000円（税抜）

申込み

詳細情報

Level 3

Zabbix認定プロフェッショナル

Level 4

Zabbix認定エキスパート

寺島 広大
CEO, Zabbix Japan LLC

現在10年以上のZabbixの経験があります。Zabbix LLCにてテクニカルサポートエンジニアとして働き、2012年10月からはZabbix日本支社のCEOとして支社の経営を行っています。2005年に設立したZabbixの日本コミュニティーの創設者でもあります。2010年には、Zabbixの日本語書籍の執筆も行っています。日本で最もZabbixの経験を持つエンジニアです。

福島 崇
NTT Communications

2009年の入社以来、Zabbixに関わる開発業務や保守サポート業務のリーダーとしてZabbixの普及に活躍している。これまでの経験で培った開発ノウハウを活かしたソースコード解析に定評がある。

福島氏がZabbixを薦める理由：オープンソースの監視ツールの中ではトップレベルの性能と、拡張性が高くさまざまな要望に対応できるツールであること。使い込むほどに監視の精度があがっていく設定の柔軟さがある素晴らしいツールであること。

トレーニング受講者の声

受講者の声

◆スピードが**早くてちょうど良い**です

◆大変お詳しい方でしたので
質問にも全てお答えいただいた。
ただ演習の見本操作が
早くついていけなかった

受講者の声

◆質問に的確に回答して頂けた。
眠くならなかった

受講者の声

◆工事音がうるさかったが
声を大きく出していただき
全て聞き取ることができました

◆**声が大きく**分かりやすい

◆**声が通っていて**聞きやすい。
疑問点にも随時答えていただけた

受講者の声

◆**ここに来ないと学べない情報**が
多くあった

◆**全て最高**

◆**日本で一番**の人に教えて
もらって幸せでした

トレーナー業 13年目です

2019年にZabbix本社から表彰されました

- 世界で1番多くのZabbix技術者を育成
  → 2019年時点で1000人超
- Alexeiに次いで世界で2番目のトレーナー
  → 現役では最古のトレーナー

Legendary trainer !

ZABBIX

docomo business

Zabbixトレーニング

~~Zabbix 6.0 認定スペシャリスト~~

Zabbix **非**認定ログスペシャリスト

Zabbix 6.0 Certified Specialist www.zabbix.com

ログ監視職人の朝は早い

「まぁ 必要に迫られて始めたログ監視ですから」

最近は**再読み込み**が起きたり
Permission Deniedが起きたりするので
気が抜けないと愚痴をこぼす。

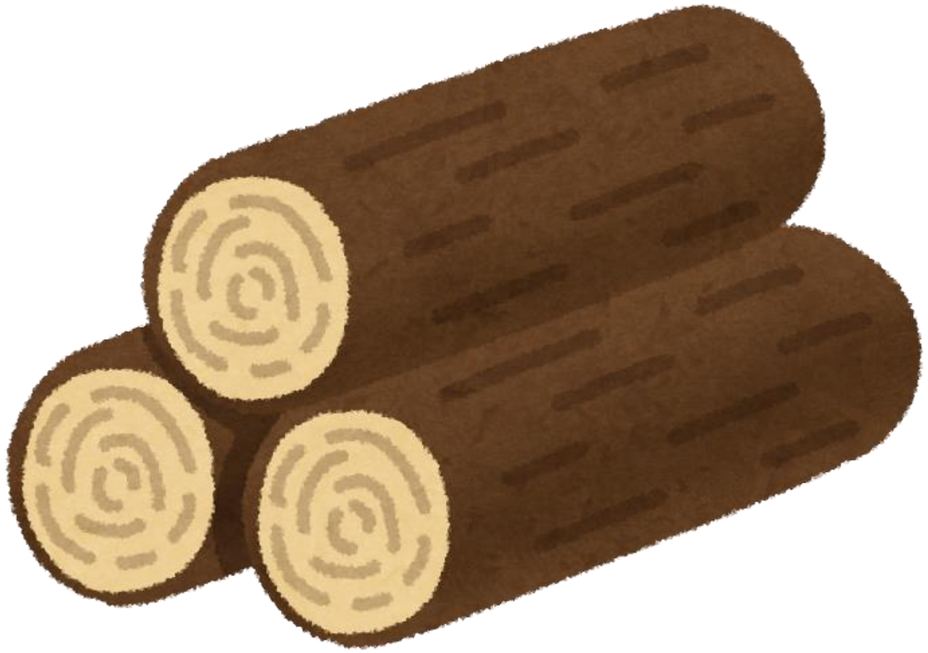
「ログはひとつひとつ温度や湿度が違う。
ここは**職人の勘の見せ所**ですよ」

オペレータルームの男たちは誰も無口だ

んなこたぁない

ログ監視は難しくない

今日は、Log監視のひみつについてを
みなさんと一緒に見ていきたいと思います。



ログローテーションの検知(logキー編)

ログローテーションの検知(logキー編)

logキーでは一体どのようにして
ログがローテーションされたことを検知するのか

チェック項目
inode番号
ファイルのMD5(最大で512バイトまで)
ファイルサイズ
ファイルの変更時間(mtime)　※ 5.0.2まで

各項目のチェック結果をエージェント内部のメモリに記憶し続けている。

ログローテーションの検知(logキー編)

チェック項目

inode番号

inode 番号は、その inode が記録されているデバイス上で**一意の整数値**である。
全てのファイルは inode に物理的にリンクされている。
プログラムがファイルをファイル名で参照するとき、システムは
そのファイル名に対応する inode を検索する。

~~民明書房刊『I know do. ～私は知っている～』より~~
ウィキペディア『inode』(https://ja.wikipedia.org/wiki/Inode)より

『inode番号が違う ≒ 違うログファイル』

ログローテーションの検知(logキー編)

チェック項目

inode番号

Windows様のファイルシステムに
inode番号なんて無いんですけど！

我がWindowsのファイルシステムはァァァァァァァアアア
世界一ィィィーーーーッ！！

ログローテーションの検知(logキー編)

チェック項目

inode番号

HaHaHa！
僕のZabbixはそのあたりも抜かりないさ。

- NTFS
  ⇒ VolumeSerialNumber ＆ 64bitのFileIndex
- ReFS
  ⇒ VolumeSerialNumber ＆ 128bitのFileId

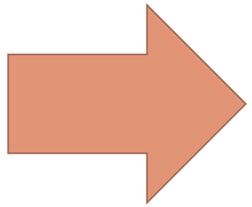
え？ FAT32やexFAT？
Best effortで64bitのFileIndexを確認してるよ
確実なチェックはできないけど頑張ってるよ

ログローテーションの検知(logキー編)

チェック項目

ファイルのMD5

2022/11/15 あいうえお
2022/11/16 かきくけこ
2022/11/17 さしすせそ
2022/11/18 たちつてと



2022/11/19 あいうえお
2022/11/20 かきくけこ
2022/11/21 さしすせそ
2022/11/22 たちつてと

MD5:
f265f1efdf167c16b390b6f3b2d3e6b3

MD5:
d2b12c8cda28d0a1cceb205916704ba7

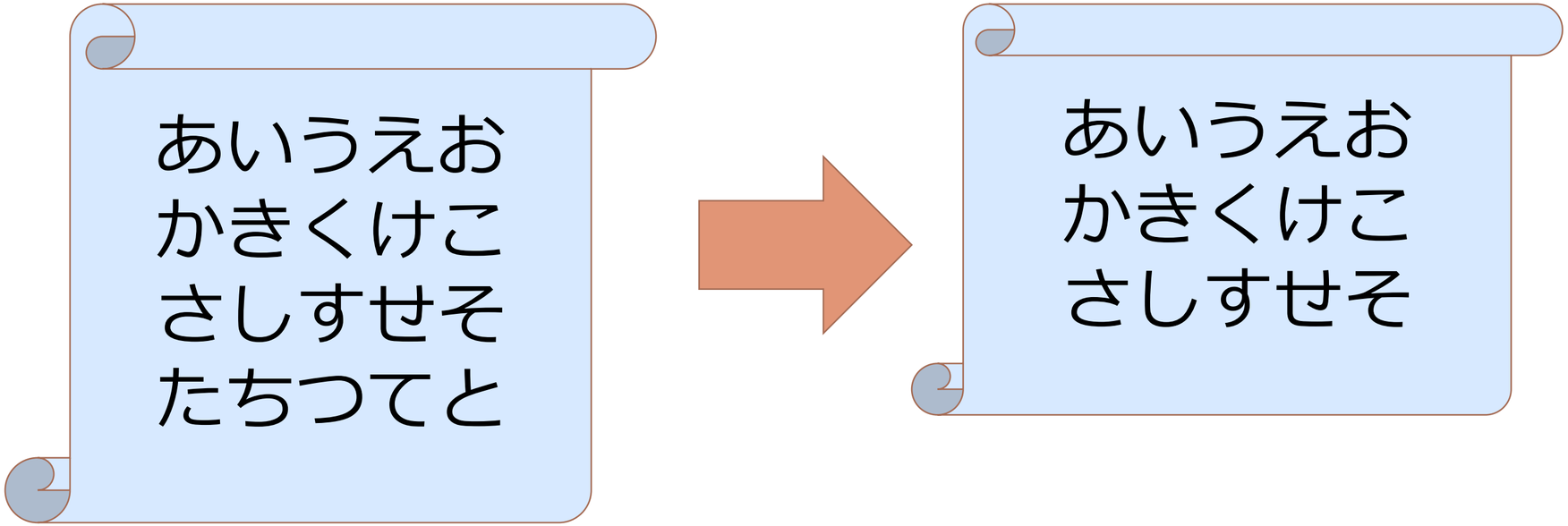
ログには普通タイムスタンプ等が含まれるから同じ文言にはならない

『MD5が違う ≒ 違うログファイル』

ログローテーションの検知(logキー編)

チェック項目

ファイルサイズ



ログと言うのは追記されるものなので

『サイズが小さくなる ≒ 違うログファイル』

という理論が成り立つ(はず)

ログローテーションの検知(logキー編)

チェック項目

ファイルの変更時間(mtime)　※ 5.0.2まで

※ ファイルサイズをチェックした際に、変化が無かった場合に確認

3種類のタイムスタンプ

atime　: time of last access (ls -lu)

mtime　: time of last **modification** (ls -l)

ctime　: time of last status change (ls -lc)

ファイルサイズが同じなのにmtimeが変化

『ファイルサイズが同じなのにmtimeが変化 ≒ 違うログファイル』

mtimeが変わるとき

あ、viエディタでログ眺めてたら
操作間違って1文字消しちゃった。てへぺろ

消した文字と同じ文字入力しなおして
念のためにセーブしておけばいいや

再読み込みが発生

mtimeが変わるとき

`$ touch /var/log/hogehoge.log` っと

あ、touchコマンドでファイル触っちゃった。
てへぺろ

再読み込みが発生

ログローテーションの検知(logキー編)

【まとめ】

ログ出力の機構以外の要素で

inode番号
ファイルの中身
ファイルサイズ
mtime

を変えてはいけません！

ログローテーションの検知(logrtキー編)

logrtキー、パラメータ多すぎ問題

logrt[file_regexp,<regexp>,<encoding>,<maxlines>,<mode>,<output>,<maxdelay>,<options>,<persistent_dir>]

パラメータ	
file_regexp	ディレクトリパス ＋ 正規表現パターン
<regexp>	正規表現によるエージェントフィルタ
<encoding>	ログファイルの文字コード
<maxlines>	秒間送信可能最大行数
<mode>	既出ログの読み飛ばしフラグ
<output>	ログ文字列の部分抜き出し位置指定
<maxdelay>	遅延可能な最大秒数
<options>	**ログファイルのローテート方式**
<persistent_dir>	「永続ファイル」のパス

『options』パラメータ

ローテート方式の二大巨頭

rotate

copytruncate

『options』パラメータはきちんと設定しよう

ローテート方式の二大巨頭

rotate

ローテート開始 → 元ファイルの → 元ファイル名で

ローテート方式とパラメータが合っていないと……

copytruncate

高確率で再読み込みが発生

ローテート開始 → 元ファイルの複製を別名で作成 → 元ファイルの中身をすべてクリアする

logrtキーの処理

logrt[/var/log/^messages(|-[0-9]{8})$]

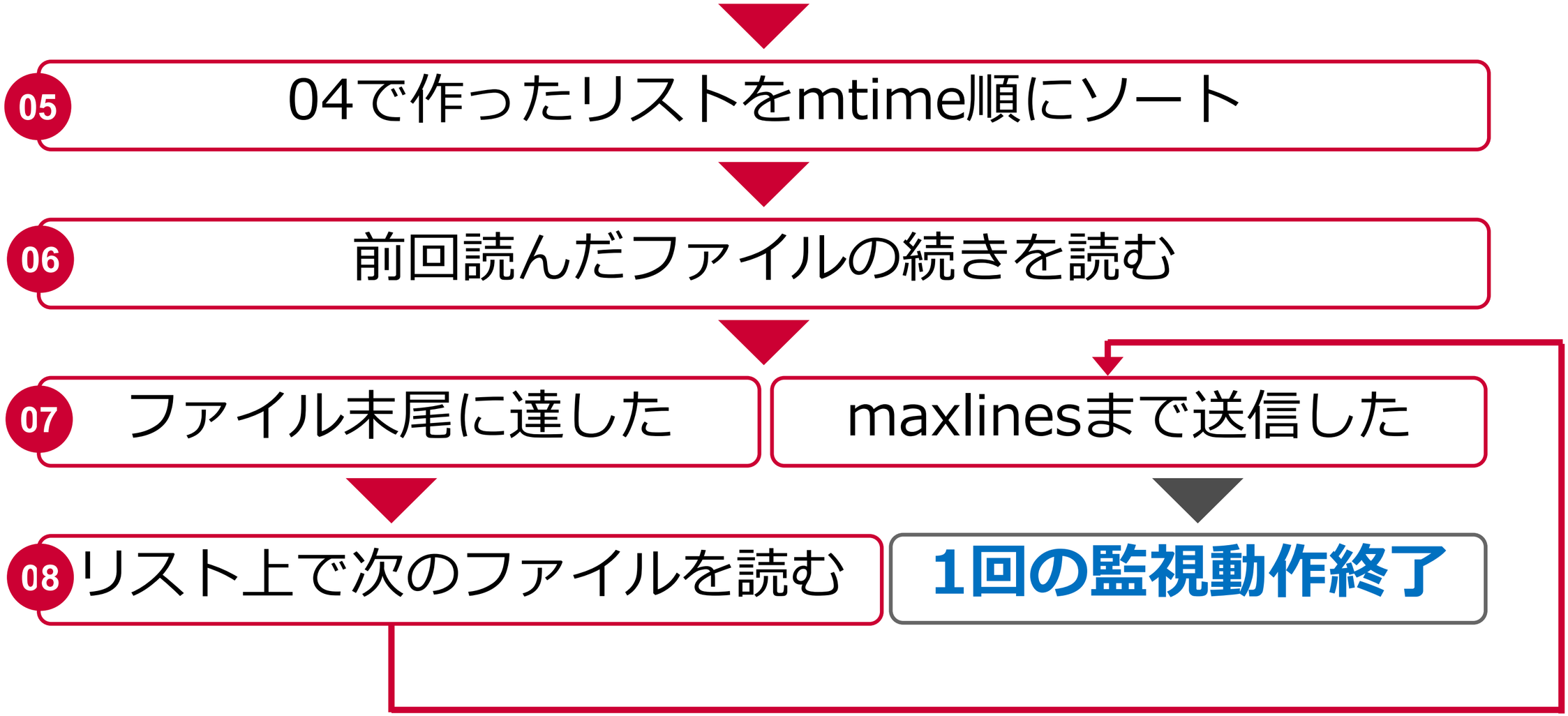
01 第1パラメータのうち、ディレクトリパスを抜き出す

02 01のディレクトリ直下へ移動

03 前回ログのmtime以上のmtimeを持つファイルが対象

04 正規表現パターンにマッチするファイルをリストアップ

logrtキーの処理



logrtキーの処理方式を踏まえて考えると

リストが崩れるようなmtimeの変更があると……

高確率で再読み込みが発生

ファイル名の正規表現が適当すぎると

```
logrt[/var/log/messages.*]
```

へんなファイルもマッチする

```
[root@ZabbixServer ~]# ls -la /var/log/
-rw-r--r-- 1 root root    16384 11月 14 13:15 .messages.swp
-rw-r--r-- 1 root root    96466 11月 14 13:11 /var/log/messages
-rw-r--r-- 1 root root 25192649 10月 26 16:50 /var/log/messages-20221026
-rw-r--r-- 1 root root   232029 10月 30 03:27 /var/log/messages-20221030
-rw-r--r-- 1 root root   480852 11月  6 03:28 /var/log/messages-20221106
-rw-r--r-- 1 root root   472848 11月 13 03:17 /var/log/messages-20221113
```

10月 26 16:50 messages-20221026
10月 30 03:27 messages-20221030
11月 6 03:28 messages-20221106
11月 13 03:17 messages-20221113
11月 14 13:11 messages
11月 14 13:15 .messages.swp

みんな大好き viエディタ

11月 13 03:17 messages-20221113
11月 14 13:11 messages
11月 14 13:15 .messages.swp

いまここまで読んだ

▼ viエディタ 閉じる

11月 13 03:17 messages-20221113
~~11月 14 13:15 .messages.swp~~
11月 14 13:11 messages

一時ファイルなので消える

ログの追記で新しくなる

ロジック的に100%正しいけど……

11月 13 03:17 messages-20221113
11月 14 13:11 messages
11月 14 13:15 .messages.swp

いまここまで読んだ

新しいログファイルだから先頭から読まないとネ！

高確率で再読み込みが発生

11月 13 03
~~11月 14 13:15 .messages.swp~~
11月 14 13:11 messages

一時ファイルなので消える

ログの追記で新しくなる

正規表現はちゃんと厳密に書こう！

logrt[/var/log/^messages(|-[0-9]{8})$]

正規表現	記号名	意味
^	ハット、Circumflex Accent	ここが行頭(開始)
()	丸かっこ、Left/Right Parenthesis	複数文字のグループ化
|	パイプ、Vertical Line	OR
[]	大かっこ、Left/Right Square Bracket	いずれか1文字にマッチする
{}	中かっこ、Left/Right Curly Bracket	直前の文字の繰り返し回数
$	ダラー、Dollar Sign	ここが行末(終了)

『messages』で始まって、何もないまま行末を迎えるか
ハイフンの後に１～9のいずれかの文字が8個続いた後に
行末を迎える文字列とマッチする！！

『前回読んだファイル』どう特定する？

Zabbixエージェントは、一度監視したものについては
作成した『正規表現にマッチするファイルリスト』と
『何番目まで処理したか』を過去1回分保持している。

01 10月 26 16:50 messages-20221026
02 10月 30 03:27 messages-20221030
03 11月 6 03:28 messages-20221106
04 11月 13 03:27 messages-20221113
05 11月 14 13:11 messages

前回処理したのは05番！

見つからない場合はリストの先頭から読む

『前回読んだファイル』どう特定する？

前回チェックしたリストのファイル名と
今回チェックしたリストのファイル名が
同じだなんて保証できないんですけど

『前回読んだ』どう特定する？(rotate編)

inode番号

ファイルサイズの減少

サイズが同じでmtimeだけ更新
→ 先頭＆末尾のMD5

mtime-reread
OR
mtime-noreread

『前回読んだ』どう特定する？(copytruncate編)

mtimeの減少

inode番号＆先頭/末尾のMD5
→ 末尾位置が異なる場合
前回の末尾位置のMD5と比較

ログローテーションの検知(logrtキー編)

【まとめ】

ローテート方式を把握しろ！

ログ出力の機構以外の要素でファイルに触れるな！

logキーと同じ点に気をつけろ！

特に中身やmtimeの編集はご法度だ！

百戦不殆(敵を知り己を知れば百戦して殆うからず)

ログ監視は知っていれば難しくない！
これでキミも「再読み込み」とお別れだ！

ご清聴ありがとうございました

docomo business

NTT Communications