株式会社エクストランス

X-MON3

X-MON fluentd 連携設定リファレンス

2021/03 版

まえがき

本書は X-MON3 系列と fluentd を連携し、より便利にご利用頂けることを目的としたマニュアルです。
そのため、基本的な LinuxOS の一般的な操作、用語などについては知識をご理解の上でお読みください。
また、本稼働中のシステムへのインストール作業などは充分に検証を行ったうえで導入するようにしてください。
いかなるシステムへの影響が発生しても、弊社は責任を負いかねますのでご了承ください。

本書執筆用に使用した環境は以下となります。
OS：CentOS 6.5 64bit 版
fluentd インストール方法：RPM 形式パッケージからインストール

なお、インストール方法が異なる場合でもファイルのパスが異なる程度で設定ファイルの書き方などは同じです。そのため、OS やインストール方法が異なる場合でも本書を参考にして頂ければと存じます。

本書以外のマニュアルについては X-MON サポートページにログインしてご確認ください。
https://x-mon.jp/support/

2014 年 07 月

改定履歴
2014 年 07 月　初版
2021 年 03 月　第 4 版

●「fluentd」sponsored by Treasure Data

●"Amazon Web Services", "AWS", "Amazon[(Chinese characters)] AWS", “Alexa Site Thumbnail”, “Alexa Top Sites”, “Alexa Web Information Service”, “Alexa Web Search”, “Amazon Athena”, “Amazon Aurora”, “Amazon Chime”, “Amazon CloudFront”, “Amazon CloudSearch”, “Amazon CloudWatch”, “Amazon Cognito”, “Amazon Connect”, “Amazon DevPay”, “Amazon DynamoDB”, “Amazon DynamoDB “Accelerator”, “Amazon EC2”, “Amazon Elastic Beanstalk”, “Amazon Elastic Compute Cloud”, “Amazon ElastiCache”, “Amazon Flexible Payments Service”, “Amazon FPS”, “Amazon Fulfillment Web Service”, “Amazon FWS”, “Amazon GameLift”, “Amazon Glacier”, “Amazon Inspector”, “Amazon Kinesis”, “Amazon Lex”, “Amazon Lightsail”, “Amazon Lumberyard”, “Amazon Machine Learning”, “Amazon Macie”, “Amazon Mechanical Turk”, “Amazon Pinpoint”, “Amazon Polly”, “Amazon Quicksight”, “Amazon RDS”, “Amazon Redshift”, “Amazon Rekognition”, “Amazon Relational Database”, “Amazon Route 53”, “Amazon S3”, “Amazon Simple Email Service”, “Amazon Simple Notification Service”, “Amazon Simple Queue Service”, “Amazon Simple Storage Service”, “Amazon SimpleDB”, “Amazon SQS”, “Amazon Virtual Private Cloud”, “Amazon VPC”, “Amazon WorkDocs”, “Amazon WorkMail”, “AWS CloudFormation”, “AWS CloudHSM”, “AWS CloudTrail”, “AWS CodeBuild”, “AWS CodeCommit”, “AWS CodeDeploy”, “AWS CodePipeline”, “AWS CodeStar”, “AWS Direct Connect”, “AWS Glue”, “AWS Greengrass”, “AWS IoT Button”, “AWS Lambda”, “AWS Marketplace”, “AWS Migration Hub”, “AWS Premium Support”, “AWS Shield”, “AWS Snowball”, “AWS Snowball Edge”, “AWS Snowmobile”, “AWS Step Functions”, “AWS X-Ray”, “CloudFront”, “DevPay”, “DynamoDB”, “EC2”, “Elasticache”, “Mechanical Turk”, “SimpleDB”, “SQS”,、ならびにその他の AWS のグラフィック、ロゴ、ページヘッダー、ボタンアイコン、スクリプト、サービス名は、米国および／またはその他 の国における、AWS の商標、登録商標またはトレードドレスです。

目次

1　用語解説

td-agent

fluentd の安定版配布パッケージです。執筆現在では RPM, deb 形式パッケージが配布されています。

fluentd の作成元である Treasure Data 社がメンテナンスを行っており、fluentd を動作させるために必要な環境（Ruby など）も一緒にインストールするため、インストール作業が非常に楽です。また、パッケージでの管理となるためバージョンアップも通常のパッケージ管理と同様に行うことができます。

タグ

fluentd に入力されたログ情報についてはタグが付与されます。ログ情報の出力時に処理の振り分けをタグにて行います。詳細については「fluentd 設定ファイルについて」の Match の項目をご参照ください。

プラグイン

fluentd ではログ情報の入出力をプラグインにて行っています。プラグインについてはインターネット上にて多数配布されていますので、ニーズに沿ったプラグインをインストールすることで様々な情報を処理することができます。

また、プラグインについては配布されているプラグインを編集することや自作することが可能です。ただ、ソフトウェアライセンスによっては編集できないプラグインが存在する可能性もございますので、編集する際には配布元にてソフトウェアライセンスをご確認ください。なお、使用しているプログラム言語は Ruby になります。

RubyGems

Ruby のサードパーティライブラリをパッケージ管理しているシステムになります。

サードパーティライブラリとは製造元及び自身以外が作成したライブラリを指します。つまり、製造元以外が作成し、インターネット上で公開されているような配布物についてはサードパーティライブラリとなります。

RubyGems はそれらをパッケージで管理するシステムとなり、インストールやアップデートについても gem コマンドを利用して簡単に行うことが出来ます。

なお、fluentd や fluentd の各入出力プラグインについても Ruby のサードパーティライブラリとなります。

2　はじめに

本書では X-MON と fluentd を連携することで、より機能を充実させるために X-MON 単体では行えない内容を実現することを目的としています。

具体的には以下のような内容が実現可能となります。

2.1 ログ監視の拡張

fluentd を利用することで X-MON で行っているログ監視をより充実させることができます。fluentd を利用したログ監視を行う上でのメリットは以下になります。

- 監視を行えるログ情報の種類を増やすことが出来る。
- 検索条件に正規表現を使用することが出来るので、より柔軟な条件設定を行える。
- 除外指定が出来るので、必要なログだけ監視することが出来る。

2.2 fluentd で取得した情報を元に監視する

fluentd で集約したログ情報を元に統計や集計を行い、それらの結果を X-MON へ送ることで新たな監視を行うことが可能となります。

具体的には以下のような例が挙げられます。

- HTTP ステータスコード別アクセス数の監視
- メールサーバに対する送信元 IP アドレス毎のアクセス数の監視
- MySQL のスロークエリの監視

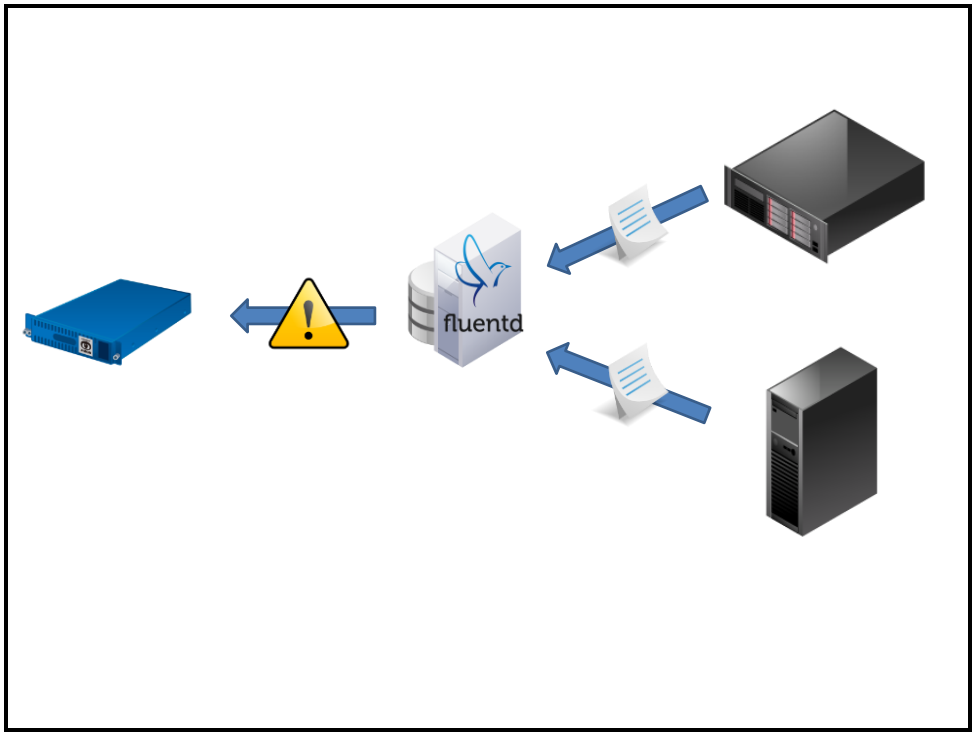


図 1 fluentd で集約したログ情報を元に監視を行うイメージ図

2.3 X-MON のログを fluentd に集約する

X-MON のログも fluentd に集約することが可能です。集約することで統計や集計処理を行うことができます。また、X-MON のイベントログ保持期間は 180 日となっておりますが、fluentd で保存することによって、さらに長い期間のログを保持することが可能となります。

具体的には以下のような例が挙げられます。

- X-MON のイベントログの蓄積
- X-MON で発生した障害件数をグラフ化
- 蓄積したログの検索

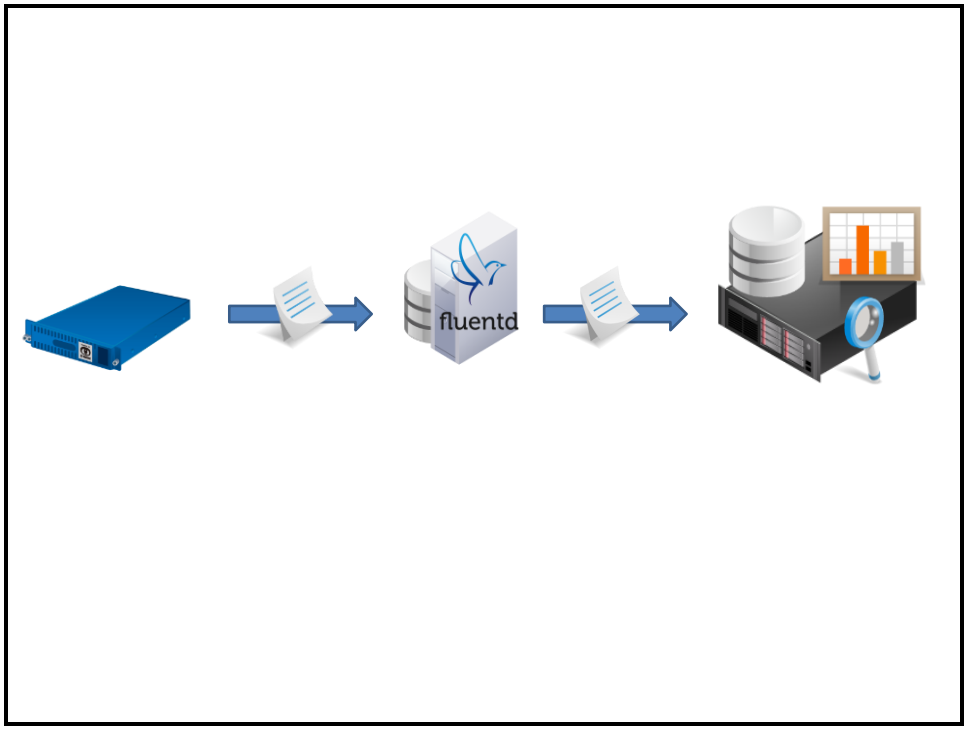
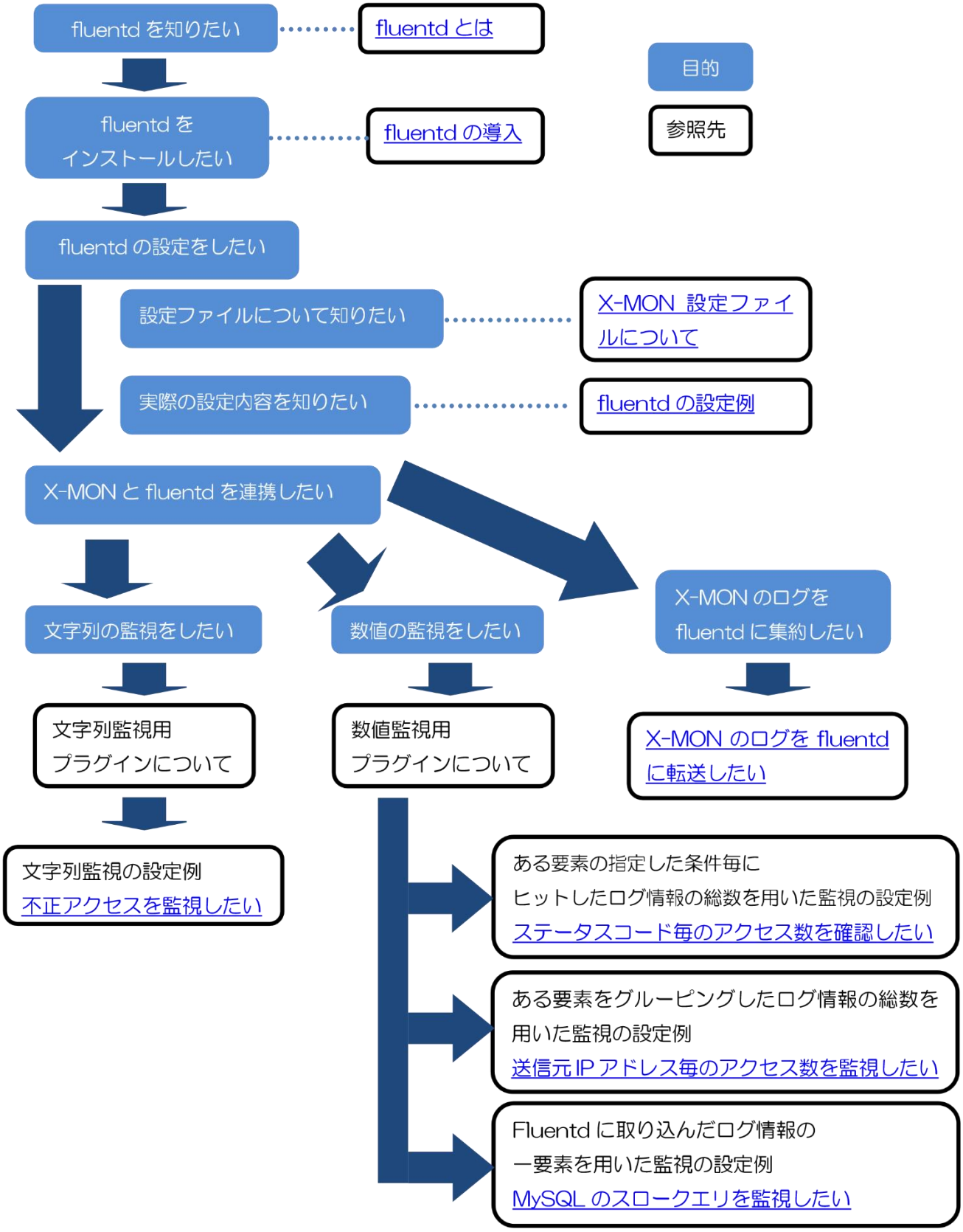


図 2 X-MON のログを fluentd にて集約するイメージ図

3　本書の参照先

本書の各ページは以下のような用途を想定しております。

目的に合ったページよりご参照頂ければと思います。



4　fluentd とは

fluentd とはログの転送や集約を行うツールです。fluentd の特徴としましては情報の入出力が柔軟であることです。

ログの転送や集約については syslog が有名ですが、syslog では扱えるログの種類が限られます。また、ログが文字列となるため、加工や処理を行うには不便な点が多いです。

fluentd はログの入出力を全てプラグインで行っています。また、そのプラグインをカスタマイズすることが可能ですので、様々な入出力に対応することができます。

入出力プラグインは多数配布されていますので、それらを組み合わせることでニーズに沿ったログの入出力を行えます。

fluentd では以下のようなログ情報を取り込むことができます。

- Apache のアクセスログ
- MySQL など DB サーバのスロークエリログ
- ネットワーク機器のシステムログ
- CloudWatch の各メトリックス値
- EC2 や RDS などの AWS 上のログ

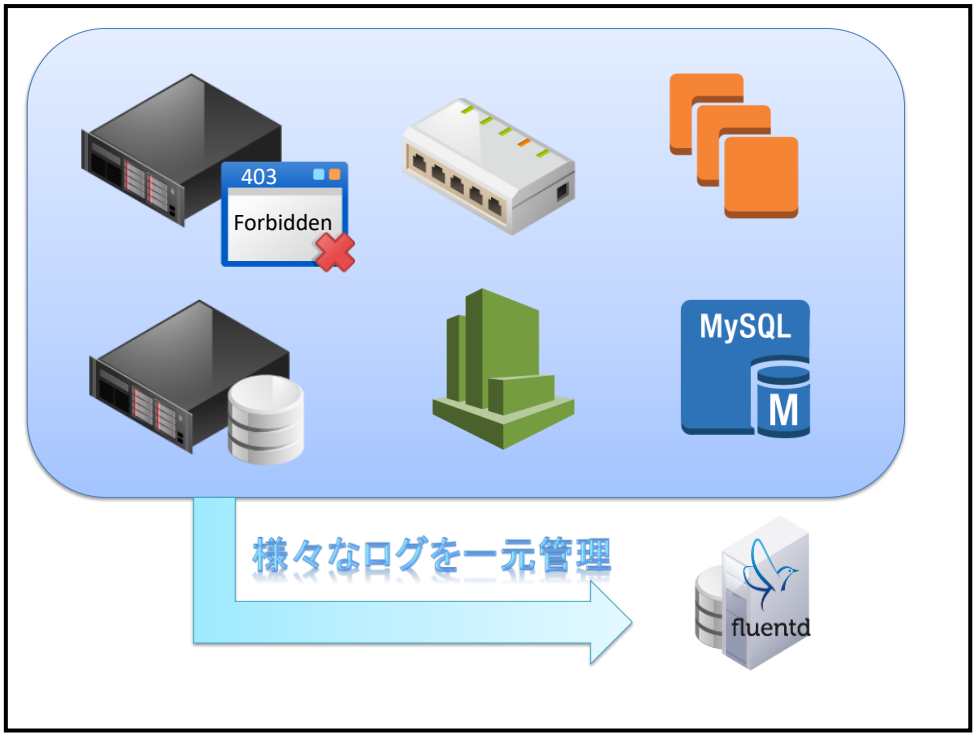


図 3 fluentd のログ情報入力イメージ図

出力につきましては以下のような出力に対応しています。

- MongoDB など DB への出力
- 検索やグラフ化を行うツールへの出力
- ファイルへの出力
- 別の fluentd サーバへの転送
- S3 への出力
- メールでの送信

また、集計や統計処理を行ってから出力することも可能です。

fluentd 自身にはログを保持する機能はありません。fluentd サーバにログを保持する場合も他の出力方法と同様にプラグインにて設定する必要があります。

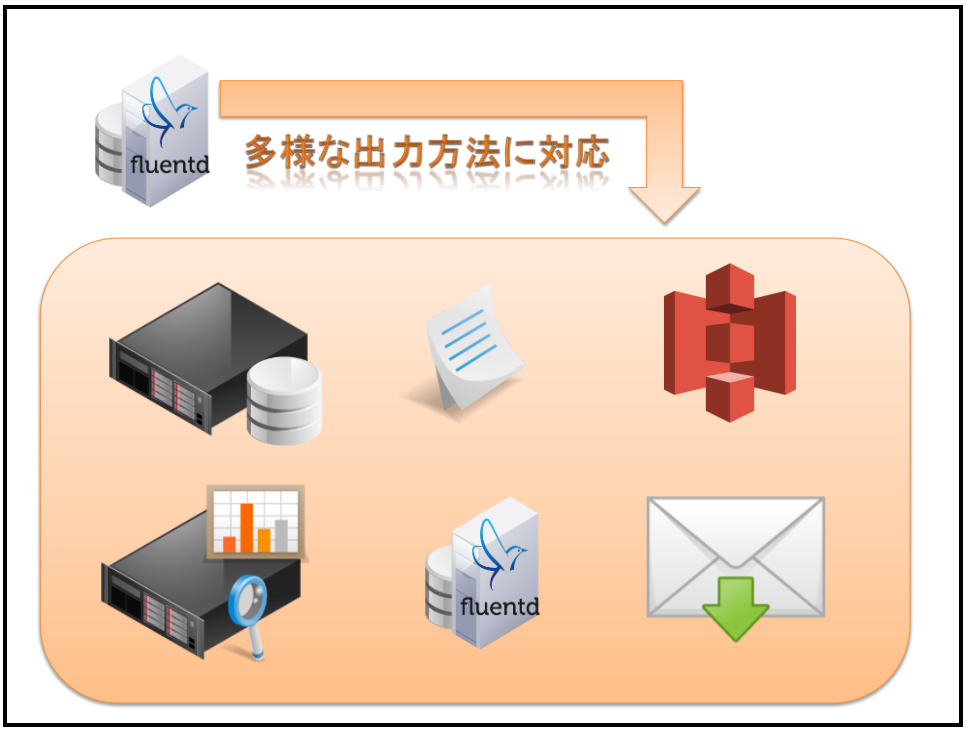


図 4 fluentd のログ情報出力イメージ図

4.1 fluentd vs 他のツール

fluentd に似たツールとしては「syslog」や「scribe」などがあげられますが、それらと比較して fluentd を使用する利点としては以下があります。

- インストールが簡単
- 拡張性が高い
- 処理がほぼリアルタイム
- Ruby で作成されているため非常に軽量
- ログの入出力をプラグインで行うため、様々なログに柔軟な対応が可能
- JSON 形式でログを扱えるため、データを扱いやすい。
- 現在もメンテナンスが行われている。

5　fluentd の導入

fluentd はいくつかのインストール方法があります。ご利用環境にあったインストール方法をお選びください。なお、執筆現在において、Windows 端末へのインストールは対応しておりません。RHEL や CentOS、Ubuntu の 10.04 や 12.04 をご利用されている場合については、別ソフトウェアのインストールを別途行う必要のない rpm / deb 形式パッケージからのインストールをお勧めします。

5.1 RPM 形式パッケージからインストール

執筆現在での対応 OS：RHEL, CentOS 5.0 以降

対応 OS 以外をご利用のお客様につきましては他の方法よりインストールを行ってください。

5.1.1　インストール用スクリプトの実行

以下のコマンドを発行することで fluentd をインストールすることが出来ます。

```
# curl -L http://toolbelt.treasuredata.com/sh/install-redhat.sh | sh
```

install-redhat.sh では新しく rpm レポジトリを登録し、yum コマンドを使用して td-agent の rpm 形式パッケージをインストールしています。

5.1.2　fluentd の実行

インストールが完了すれば以下のコマンドにて fluentd を起動することができます。

```
# /etc/init.d/td-agent start
```

5.2 deb パッケージからインストール

執筆現在での対応 OS：Ubuntu 12.04 LTS / Precise, Ubuntu 10.04 LTS / Lucid

対応 OS 以外をご利用のお客様につきましては他の方法よりインストールを行ってください。

5.2.1　インストール用スクリプトの実行

以下のコマンドを発行することで fluentd をインストールすることが出来ます。

Ubuntu Precise の場合

```
# curl -L http://toolbelt.treasuredata.com/sh/install-ubuntu-precise.sh | sh
```

Ubuntu Lucid の場合

```
# curl -L http://toolbelt.treasuredata.com/sh/install-ubuntu-lucid.sh | sh
```

install-ubuntu-precise.sh / install-ubuntu-lucid.sh では新しく apt レポジトリを登録し、apt-get コマンドを使用して fluentd の deb 形式パッケージをインストールしています。

5.2.2 fluentd の実行

インストールが完了すれば以下のコマンドにて fluentd を起動することができます。

```
# /etc/init.d/td-agent start
```

5.3 RubyGems からインストール

fluentd についても RubyGems で配布されておりますので、gem コマンドを使用してインストールすることが可能です。

5.3.1 Ruby のインストール

Gem コマンドを使用するには Ruby(1.9.2 以上)をインストールする必要があります。

Ruby のインストールについては公式 HP をご参照ください。

https://www.ruby-lang.org/

- ダウンロードページにインストール方法が記載されています。

5.3.2 fluentd のインストール

以下のコマンドを発行することで fluentd がインストールされます。

```
# gem install fluentd
```

5.3.3 fluentd の実行

インストールが完了すれば以下のコマンドにて fluentd を起動することができます。

```
# fluentd --setup ./fluent ※ 初回起動時のみ
# fluentd -c ./fluent/fluent.conf -vv &
```

※RubyGems からインストールした場合に起動スクリプトは作られません。

6　fluentd の設定

fluentd の設定についてご紹介します。

6.1 fluentd の設定ファイルについて

fluentd の設定ファイルについてはインストール環境によって異なります。RPM / deb 形式パッケージでインストールされた場合は「/etc/td-agent/td-agent.conf」が設定ファイルとなります。RubyGems からインストールされた場合につきましては、fluentd 起動時に「-c」オプションで指定したファイルが設定ファイルとなります。設定ファイルにはサンプルの設定が入っていますので、そちらもご参照ください。

fluentd の設定ファイルについては以下の 3 つの要素から構成されています。

- Include
  - 他の fluentd 設定ファイルを読み込む設定を記載する要素
- Source
  - fluentd へログ情報を取り込む設定を記載する要素
- match
  - fluentd からログを出力する設定を記載する要素

6.1.1　include

他の fluentd 設定ファイルを読み込みます。読み込むファイルの指定については以下が可能です。

```
# 絶対パス及び相対パスでの指定が可能です。
include /path/to/fluentd.conf
include ex_fluentd.conf

# 正規表現を用いた設定が可能です。
include fluentd/*.conf

# HTTP 経由で接続できるファイルを指定することができます。
include http://example.com/fluent.conf
```

6.1.2 source

ログ情報の入力方法を設定する要素となりますので、入力プラグインを指定します。

例えば、別の fluentd から転送されてきたログを取り込む場合には以下の設定となります。

```
<source>
  type forward
  port 24224
</source>
```

設定する値については各プラグインで異なりますが、使用するプラグインの設定については「type」項目に記載します。記載する内容は各プラグインの接頭語を除いた名称となります。プラグインについては後述をご参照ください。

6.1.3 match

ログ情報の出力先を設定する要素となりますので、出力プラグインを指定します。指定したタグの条件にマッチしたログに対して、記載した内容が実行されます。

同じ条件を設定した match 要素が複数ある場合には設定ファイルの上部で記載した match 要素のみが実行されますのでご注意ください。同じ条件で異なる処理を通す場合には「copy」や「forest」プラグインをご利用ください。

例えば、別の fluentd へログを転送する場合には以下のような設定となります。

```
<match xmon.sample>
  type forward
  <server>
    host 192.168.100.50
    port 24224
  </server>
</match>
```

以上の設定で、xmon.sample というタグの付与されたログについては 192.168.100.50 のサーバに転送されます。

指定する条件については、以下の表現を使用することができます。

- *
  - 一つの任意のタグに合致

    例：x-mon.*

    ヒットするタグ：x-mon.log, x-mon.alert, x-mon.service

    ヒットしないタグ：x-mon, x-mon.log.event

- **
  - 0 個以上のタグに合致
    例：x-mon.**
    ヒットするタグ：x-mon, x-mon.log, x-mon.log.event
    ヒットしないタグ：x-mon2.sample, sample.x-mon
- {A, B, C}（※ A,B,C はそれぞれ条件）
  - A,B,C いずれかの条件に合致
    例：x-mon.{critical.service, critical.host}
    ヒットするタグ：x-mon.critical.service, x-mon.critical.host
    ヒットしないタグ：x-mon.ok.service, x-mon.critical

6.2 プラグイン

fluentd は前述した通り、ログ情報の入出力をプラグインで行っています。また、プラグインについてはインターネット上で多数配布されています。
このプラグインを組み合わせることにより様々なログ収集や保存先の変更、統計処理などが容易に行えます。

6.2.1 プラグインのインストール

fluentd プラグインのインストール方法については、下記の二つがあります。

6.2.1.1 gem コマンドを使用したインストール

RubyGems に公開されているプラグインについては gem コマンドでインストールが可能です。
gem コマンドでダウンロード可能なプラグインについては以下のコマンドにてインストールが行えます。

```
# /usr/lib64/fluent/ruby/bin/gem install {プラグイン名}
```

※実行コマンドのパスについては環境により異なります。

6.2.1.2 直接ファイルを設置する方法

RubyGems で配布されていないプラグインや自作したプラグインはプラグインディレクトリ直下に置き、fluentd を再起動することでプラグインを利用することが可能となります。
RPM / deb 形式パッケージでインストールした場合は「/etc/td-agent/plugin」が対象のディレクトリとなります。
RubyGems やソースコードからインストールした場合は「fluentd --setup」コマンドで指定したディレクトリ直下に存在する「plugin」ディレクトリが対象のディレクトリとなります。

6.2.2 入力プラグイン

fluentd にログを吸い上げるプラグインです。入力プラグインをカスタマイズすることで、様々な情報を fluentd に取り入れることが可能となります。また、取り入れたログについては全てにタグが割り振られます。プラグインの接頭語として「in_」が付与されます。

代表的な入力プラグインには以下があります。

- in_forward
  - 他の fluentd サーバより転送されてきたログ情報を取り入れます。
- in_tail
  - ファイルに文字が出力されれば、それをログ情報として取り入れます。
- in_syslog
  - fluentd サーバに転送された syslog をログ情報として取り入れます。
- in_mysql_slow_query
  - MySQL のスロークエリログを取得し、ログ情報として取り入れます。

6.2.3 出力プラグイン

fluentd に入力されたログをファイルに書き出したり、別のサーバへ転送したりするプラグインです。単純に出力するだけでなく、集計などを行った結果を出力するプラグインもあります。プラグインの接頭語として「out_」が付与されます。

代表的な出力プラグインには以下があります。

- out_forward
  - 他の fluentd サーバにログを転送します。転送したログについては in_foward プラグインで受け取ります。
- out_file
  - 指定したファイルにログ結果を出力します。
- out_exec
  - ログ情報を引数として渡し、プログラムを実行します。実行するプログラムについては言語を問いません。
- out_copy
  - 同じログを複数のプラグインへ渡すことができます。fluentd では先に条件にマッチした内容を実行するため、同じログを複数のプラグインで実行するためには out_copy プラグインを利用する必要があります。
- out_s3
  - ログの結果を AWS 上の S3 に転送します。

- out_datacounter
  - ログの結果より、指定した要素の値のうち正規表現でマッチしたログの数をカウントします。

6.2.4 バッファプラグイン

バッファリングを行うプラグインです。バッファリングを行うことにより、出力プラグインが出力に失敗しても再送処理が可能となるため、情報の紛失を防ぐことができます。プラグインの接頭語として「buf_」が付与されます。

代表的なバッファプラグインは以下があります。

- buf_memory
  - メモリ上にてバッファリングを行います。
- buf_file
  - ファイル上にてバッファリングを行います。

6.3 fluentd の設定例

次に、実際の設定例を基にいくつか代表的なプラグインの設定内容を説明致します。

なお、以降に記載する設定につきましてはX-MONに特化した内容ではございませんが、X-MONのログに関しても利用できる内容です。

6.3.1 取り込んだログ情報をファイルに出力する

この設定につきましては、ログ情報を保存するという意図もございますが、実際に取り込んだログ情報がどのような値なのか確認をする作業にも利用できます。

6.3.1.1 ファイルに出力する設定（デバッグ出力方法）

ファイルに出力するには「out_file」プラグインを使用します。

「out_file」プラグインは fluentd に標準で同梱されているため、インストール作業は不要です。

Fluentd の設定ファイルに以下を記載します。

```
<match debug.**>
  type file
  path /tmp/debug.log
</match>
```

この設定では「/tmp/debug.log.**********（*はランダムな半角英数字）」にタグ名が「debug」から始まるログ情報が出力されます。

なお、出力されるファイルについてはログが出力される度にファイル名称が同じでも一度ファイルを再生成しているため、tail コマンドでファイルを出力させていても新たに追加されたログ情報が表示されませんのでご注意ください。

出力例は以下の通りです。（表示の都合上改行をしておりますが、実際は一行です。）

```
2014-03-30T16:07:03+09:00
 x-mon.maillog_to
{"host":"localhost","ident":"postfix/local","pid":"11091","messageid":"11DE6207E7","to":"root@test.x-mon.local","message":"relay=local,delay=0.01,delays=0/0/0/0,dsn=5.2.2,status=bounced(cannot update mailbox /var/mail/root for user root. error writing message: File too large)"}
```

出力された情報はそれぞれ「ログ情報の出力時間 タグ名 ログ情報」となります。

Fluentd の設定をされる際に、どのような値が取得できているのかをご確認される際はこの設定をご参考にしてください。

6.3.2 syslog を fluentd サーバに取り込む

複数ホストのログを管理する場合に、各サーバのログを閲覧するには非常に労力がかかります。ログを一台のサーバに集約することでその負荷を軽減することができます。syslog については直接 fluentd サーバにログを送ることができます。

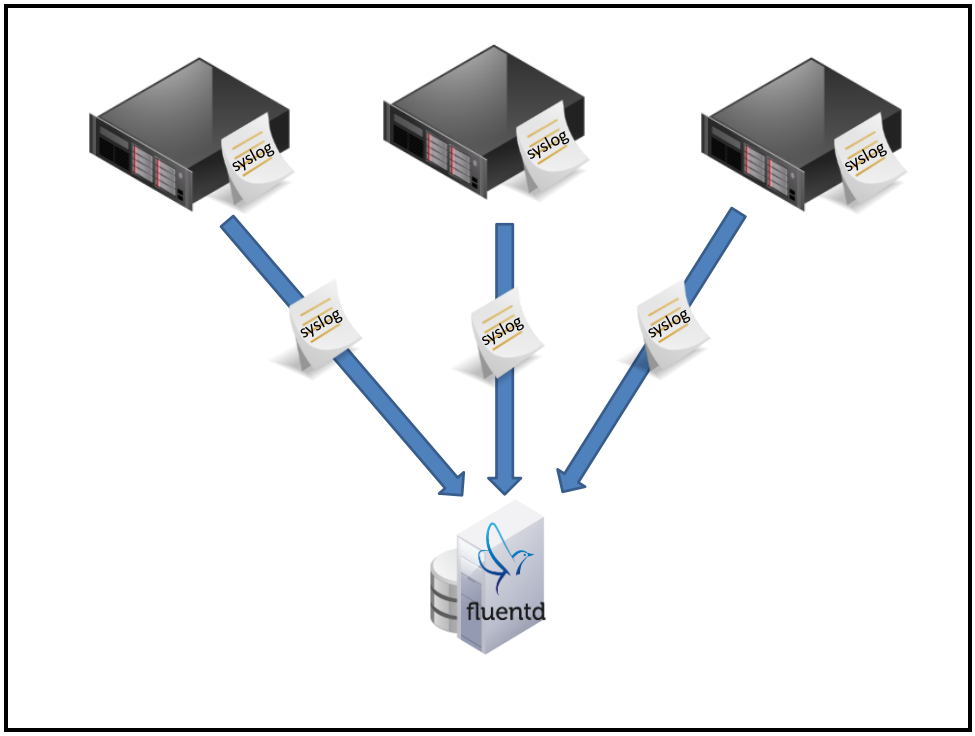


図 5 syslog の転送イメージ図

6.3.2.1 転送元の設定

転送元の設定としては syslog を fluentd サーバへ転送する設定が必要となります。

例として以下の設定を syslog の設定ファイルに行います。

```
*.* @192.168.100.50:5140
```

なお、fluentd で syslog を取り込むためには UDP で通信を行う必要があります。そのために、IP アドレス前にある「@」が 1 つであることに注意してください。「@」を 2 つ記載すると TCP で通信を行います。

6.3.2.2 fluentd サーバでの設定

転送されてきた syslog を受信するためには「in_syslog」プラグインを使用します。

fluentd の設定ファイルに以下を記載します。

```
<source>
  type syslog
  port 5140
  bind 0.0.0.0
  tag system
</source>
```

各設定内容は以下の通りです。

- port
  - syslog を受信するポート。
  - syslog 転送設定時に記載した「:(コロン)」以降と合わせる必要があります。
- bind
  - 受信を許可するサーバを指定します。
- tag
  - 取得したログ情報に付与されるタグの先頭部分を指定します。
  - 実際に付与されるタグについては「{tag で指定した文字列}.{ファシリティ}.{プライオリティ}」となります。

6.3.3 apache のアクセスログを fluentd サーバに取り込む

「in_tail」プラグインを使用することでファイルに追記された情報をログ情報として扱うことが出来ます。

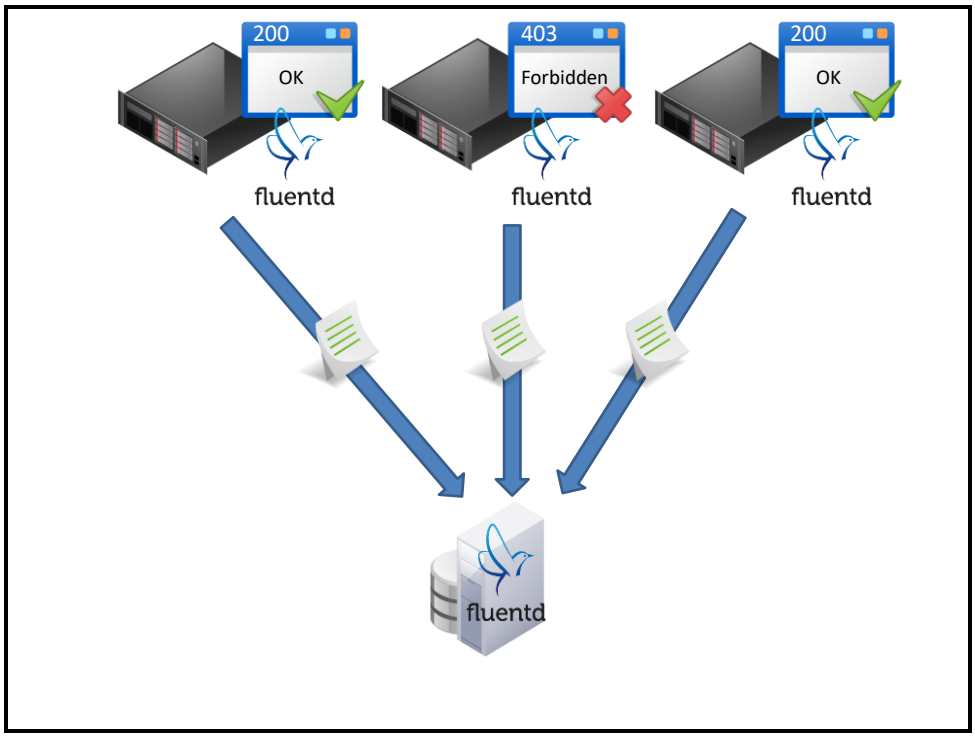


図 6 アクセスログ転送のイメージ図

6.3.3.1 アクセスログファイルからログ情報を fluentd に取り入れる設定

アクセスログを fluentd サーバへ転送するためには以下の条件を整えていることが前提となります。

- fluentd がインストールされている
- アクセスログファイル及びディレクトリに fluentd を実行しているユーザの実行権限が付与されている

前提条件を整えた上で fluentd の設定ファイルに以下を記載します。

```
<source>
  type tail
  path /var/log/httpd/access_log
  pos_file /var/log/td-agent/access_log.pos
  tag www1.apache.access
  format apache2
</source>
```

各設定内容は以下の通りです。

- path
  - 読み込むファイルを指定します。
- pos_file
  - 最後に読み込んだファイルの場所を記録します。
- tag
  - ログ情報に付与するタグ名を付与します。
- format
  - 取り込むログ情報のフォーマットを指定します。正規表現での指定も可能ですが、特定のログについては既にフォーマットが指定されているものがあります。Apache のログについてもフォーマットが指定されていますので、正規表現で指定する必要はありません。既に用意されているフォーマットとしては以下がございます。
    - apache
    - apache2
    - syslog
    - nginx

実際に取得したログ情報は以下の通りです。

```
{
  "host":"192.168.13.72",
  "user":null,
  "method":"GET",
  "path":"/",
  "code":200,
  "size":14447,
  "referer":null,
  "agent":"Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36 (KHTML, like Gecko)
  Chrome/33.0.1750.146 Safari/537.36"
}
```

6.3.3.2 取り込んだログ情報を fluentd サーバへ転送する設定

取り込んだログ情報を別の fluentd サーバに転送するにはログ転送元サーバで以下の設定を行います。

```
<match apache.access>
type forward
<server>
  name fluentdServer
  host 192.168.100.50
  port 24224
</server>
<secondary>
  type file
  path /var/log/td-agent/forward-failed
</secondary>
</match>
```

各設定内容は以下の通りです。

- <server>要素
  - <server>要素は必ず一つは必要となります。複数の server を設定することで負荷分散及び冗長構成を組み立てることが可能です。
    - name
      - サーバの名称を入力します。ここに入力された値は送信失敗時などのエラーメッセージに使用されますので、分かりやすい名称を設定してください。
    - server (必須)
      - 転送先 fluentd サーバの IP アドレスまたはホスト名を記載します。
    - port
      - ログ転送に使用するポート番号です。デフォルトは 24224 となります。
- <secondary>要素
  - <server>要素全ての通信に失敗した場合に行う処理を記載します。記載のしかたについては通常の<match>要素と同じです。
  - 今回の例では/var/log/td-agent/forward-failed にログを書きだすように設定しています。

6.3.3.3 転送されてきたログを受信する設定

ログ転送先サーバの fluentd にて受信する際には「forward」プラグインを使用します。

設定については以下の通りです。

```
<source>
  type forward
</source>
```

これで他の fluentd から転送されてきたログ情報を取り込むことが出来ます。取り込んだログ情報のタグについては転送時に付与されているタグがそのまま付与された状態になっています。

6.3.4 Windows のログを fluentd に取り込みたい

fluentd は Windows サーバへの導入は行えない、と先述しましたが、Windows のログを fluentd に集約できない、ということはございません。ただし、その前提と致しましては Windows サーバのログを syslog に変換して、Linux サーバへ転送する必要があります。Windows のログを syslog として扱えるようにするソフトウェアとしましては「NTsyslog」や「nxlog」などがあります。これらのインストールや設定方法については各公式 HP などをご覧ください。

Windows サーバのログを fluentd サーバへ転送することが出来ましたら、後は設定例で既に紹介しているように syslog を fluentd に取り込む設定を行えば、fluentd にて windows のイベントログを集約することが可能です。

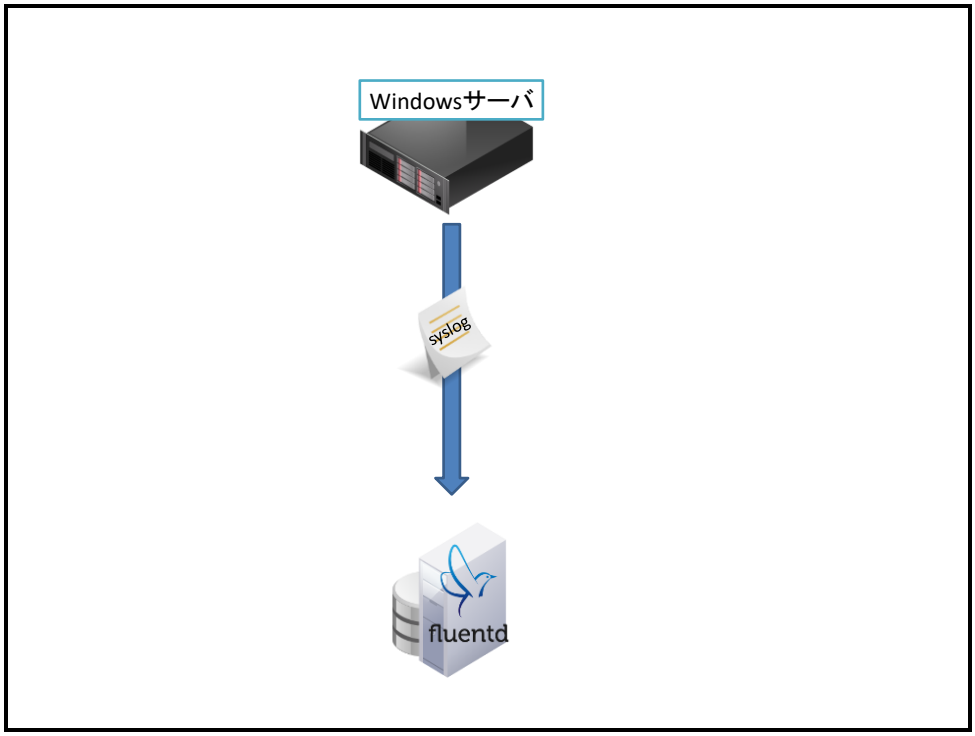


図 7 Windows サーバより fluentd に syslog 経由で転送するイメージ図

6.3.5 CloudWatch の値を取得したい

AWS の CloudWatch では値を 2 週間しか保持することができませんので、長期間 CloudWatch の結果を保持するためには別途保存処理を行う必要があります。API やコマンドラインツールなど CloudWatch の値を外部から取得する方法はいくつかありますが、それ用にプログラムを作成したりするのは手間です。しかし、fluentd ではプラグイン一つをインストールすることで CloudWatch の値を取得することが可能です。

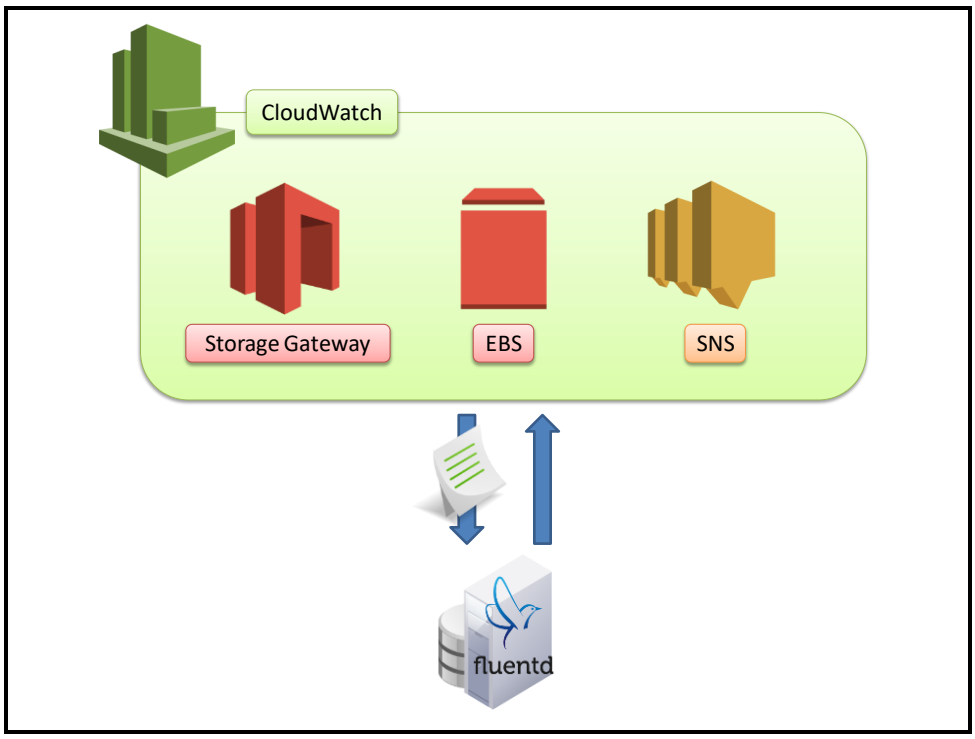


図 8 CloudWatch を用いて値を取得するイメージ図

6.3.5.1 必要なプラグインのインストール

CloudWatch から値を取り入れるには「cloudwatch」プラグインを使用します。RubyGems にて配布されておりますので、以下のコマンドにてインストールすることが出来ます。

```
# /usr/lib64/fluent/ruby/bin/gem install fluent-plugin-cloudwatch
```

※実行コマンドのパスについては環境により異なります。

6.3.5.2 CloudWatch の値を fluentd に取り込む

```
<source>
  type cloudwatch
  tag cloudwatch
```

```
  aws_key_id  YOUR_AWS_ACCESS_KEY
  aws_sec_key YOUR_AWS_SECRET_KEY
  cw_endpoint CloudWatch ENDPOINT_NAME

  namespace  AWS/EC2
  statistics  AVERAGE
  metric_name CPUUtilization, NetworkIn, NetworkOut
  dimensions_name  InstanceId
  dimensions_value TARGET_INSTANCE_ID
  period 300
  interval 300
</source>
```

各設定項目は以下の通りです。

- tag
  - 取り込んだログ情報に付与するタグ名
- aws_key_id
  - AWS のアクセスキー
- aws_sec_key
  - AWS のシークレットキー
- cw_endpoint
  - CloudWatch のエンドポイント名を指定します。エンドポイント名はリージョンにより異なります。
- namespace
  - 監視対象のサービス名を指定します。
- statistics
  - 取得する値の算出方法を指定します。
- metric_name
  - 取得する監視の要素名を指定します。カンマ区切りで複数指定することができます。
- dimensions_name
  - 識別子が入っている要素名を指定します。
- dimensions_value:
  - 監視対象の識別子を指定します。

取得したログ情報は以下の通りです。

```
{ "CPUUtilization":0.662}
  {"NetworkIn":381.0}
{"NetworkOut":303.4}
```

このプラグインでは CloudWatch から取得してきた情報はメトリック毎にレコードが分かれます。
そのため、今回の設定では 3 レコードの情報を取得します。

6.3.6　ログを S3 に保存したい

S3 へログを保存する際は「out_s3」プラグインを使用します。

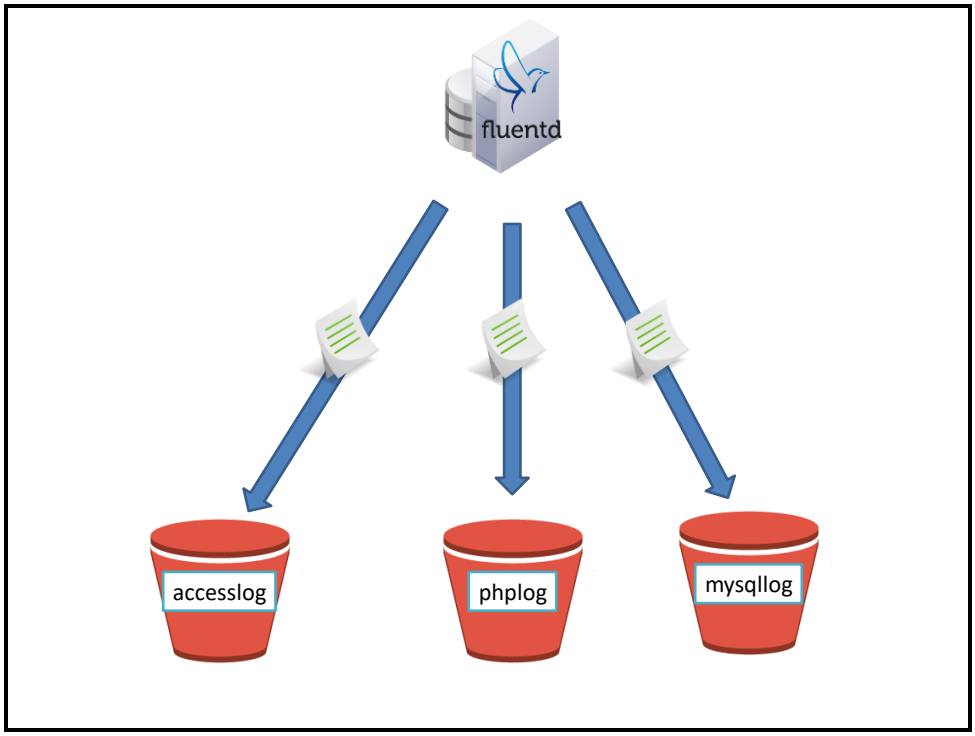


図 9 fluentd のログを S3 に保存するイメージ図

6.3.7　fluentd のログを s3 へ転送する設定

fluentd のログを転送するためには以下の条件を整えていることが前提となります。

- S3 のバケットが用意されている

また、以下の情報を用意する必要があります。

- S3 へのアップデート権限を持つユーザのアクセスキー及びシークレットキー
- データをアップロードする S3 のエンドポイント名

前提条件を整えた上で fluentd の設定ファイルに以下を記載します。

```
<match s3.**>
    type s3
    aws_key_id YOUR_AWS_ACCESS_KEY
    aws_sec_key YOUR_AWS_SECRET_KEY
    s3_endpoint S3_ENDPOINT_NAME
    s3_bucket accesslog
    path logs/
    buffer_path /var/log/td-agent/s3
    store_as txt
    time_slice_format %Y%m%d%H%M%S
    time_slice_wait 10m
</match>
```

各設定項目は以下の通りです。

- aws_key_id
  - AWS のアクセスキー
- aws_sec_key
  - AWS のシークレットキー
- s3_endpoint
  - S3 のエンドポイント名を指定します。
  - エンドポイント名はリージョンにより異なります。
- s3_bucket
  - ログファイルを保存するバケット名を指定します。
- path
  - ログファイルを保存するバケット内のパスを指定します。
- buffer_path
  - ログを S3 へ出力するまでに保存するファイルのパスを指定します。
- time_slice_format
  - ログファイルを出力する際に使用する時刻表記の形式を指定します。
- time_slice_wait
  - ログファイルを保存する間隔を指定します。今回の設定では 10 分毎にログファイルを保存するようになっています。
- store_as

- ログファイルを S3 に保存する際の拡張子を設定します。
- 設定できる拡張子には以下があります。
  - gzip：デフォルト
  - json
  - txt
  - lzo

6.3.8 ログを MongoDB へ保存したい

fluentd は大量のログを集約する目的で利用されております。そのため、fluentd で集約したログ情報を DB に保持する際は、スケーラブルな MongoDB を採用するケースが多いです。
MongoDB へ出力するためには「mongo」プラグインを使用します。

なお、設定を行うためには以下の条件を整えていることが前提となります。

- ログ情報を保存するサーバに MongoDB が既にインストールされている

以下が実際の設定例です。
なお、例ではログ情報を保存するサーバを 192.168.100.51 としています。

```
<match mongo.**>
type mongo
host 192.168.100.51
port 27017

database fluentd
collection fluentdLog

flush_interval 10s
</match>
```

各設定項目は以下の通りです。

- Host
  - データを保存する MongoDB が設定されているサーバの IP アドレスまたはホスト名を指定します。
- Port
  - MongoDB が使用しているポート番号を指定します。MongoDB がデフォルトで使用しているポート番号は 27017 です。

- database, collection
  - データを格納するデータベース名及びコレクション名を指定します。
- flush_interval
  - データをプラグインに流し込む間隔を指定します。デフォルトでは 60s が指定されています。単位としては s(秒), m(分), h(時間)を使用することが出来ます。

7　X-MON と fluentd の連携について

最後に X-MON と fluentd の連携についてご紹介します。

7.1 X-MON への出力プラグイン

X-MON への出力プラグインについては以下の 2 種類があります。

- 文字列監視用プラグイン
- 数値監視用プラグイン

文字列監視用プラグインにつきましてはログ監視を行う際に利用します。検索対象の文字を正規表現で指定することや除外対象を指定することもできます。
数値監視用プラグインにつきましては取得したログ情報内にある数値や集計を行った後の情報から監視を行うことが出来ます。また、グラフの作成にも対応しております。

7.1.1　文字列監視用プラグイン

文字列監視用プラグインは X-MON のログ監視を更に充実させるために使用します。検索条件や除外条件を正規表現で指定することが出来ますので、細かな監視設定が可能となります。

7.1.1.1　　設定内容

文字列監視用プラグインの設定項目は以下の通りです。

```
  <match xmon.log.**>
  type xmon_log
  xmon 192.168.100.51
  host BackupServer
  service DailyBackup
  regexp1 message DAILY BACKUP
  regexp2 status Failed
</match>
```

各設定項目は以下の通りです。

- xmon (必須)
  - 結果を送信する X-MON サーバの IP アドレスまたはホスト名を指定します。ホスト名を指定する場合はそのホスト名が名前解決出来る必要があります。

- host (必須)
  - 監視結果を通知するサービスのホスト ID を指定します。
- service
  - 監視結果を通知するサービスのサービス ID を指定します。
- status
  - X-MON に通知する際のステータスを設定します。入力できる値は「OK」、「CRITICAL」、「WARNING」、「UNKNOWN」となります。デフォルトでは「CRITICAL」が設定されています。
  - 障害ログと対になる復旧ログが分かっている場合につきましては、この項目を「OK」と設定しておくことで、復旧ログを検知すると自動で復旧させることができます。なお、それ以外のログにつきましてはパッシブチェックの結果送信を用いて手動で「OK」としてください。
- target
  - X-MON に表示されるステータス情報に表示するログの要素名を指定します。カンマ区切りで複数指定することが可能です。デフォルトでは全要素が表示されます。
- regexp[1-20]
  - 検索条件を指定します。
  - 記載方法については左から半角スペース区切りで「regexpN 要素名 検索条件」(N:1 から 20 の数字)となります。検索条件については Ruby の「Regexp」クラスで使用できる正規表現が使用できます。正規表現の説明につきましては、後述します。
  - また、一つも条件を記載しなかった場合は全てのログを検索条件に合致したものと認識します。
- exclude[1-20]
  - 検索結果から除外する条件を指定します。
  - 記載方法については regexp と同様です。

7.1.2 数値監視用プラグイン

数値監視用プラグインではログ情報内にある数値や統計や集計を行った結果より監視を行います。X-MON では行えない監視もこのプラグインを使用することで可能となります。また、グラフの作成にも対応しておりますので統計などをグラフに残したい場合にも利用が可能です。

7.1.2.1 設定内容

数値監視用プラグインの設定項目は以下の通りです。

```
<match xmon.monitoring.**>
type xmon
xmon 192.168.10.51
host WebSever
service STATUS_CODE
description_name_type fixed
description_name 2:xx,3_xx,4;xx,5xx
keys 2xx_rate,3xx_rate,4xx_rate,5xx_rate
</match>
```

- xmon (必須)
  - 結果を送信する X-MON サーバの IP アドレスまたはホスト名を指定します。ホスト名を指定する場合はそのホスト名が名前解決出来る必要があります。
- host (必須)
  - 監視結果を通知するサービスのホスト ID を指定します。
- service
  - 監視結果を通知するサービスのサービス ID を指定します。
- keys
  - 監視に用いる値が格納されている要素名を指定します。指定する要素が配列となっている場合については「.(ドット)」区切りで要素名を選択することで配列の中の要素も指定することが可能です。
  - また、レコード数が不定の場合については「*(アスタリスク)」を用いることで対応することが出来ます。例えば「key1.*.key2」と指定した場合については「key1」という配列の中にある全レコードの「key2」要素の値を使用するということになります。実際の設定例については後述します。
- description_name_type
  - 値を識別する名称の設定方法を指定します。設定方法には以下があります。
    - Number … 取得した順に「Value1」「Value2」とする。
    - Fixed … description_name で指定した名称を設定する。
    - Auto … description_name で指定した要素の値を識別子として設定する

- description_name
  - 値を識別する名称を設定します。Description_name_type が「fixed」または「auto」の場合は必須項目となります。
  - 入力する内容については「fixed」の場合は実際に設定する名称を入力します。「auto」の場合は識別子として使用する値が格納されている要素名を指定します。カンマ区切りで複数指定することが可能です。
- warning, critical
  - それぞれ WARNING, CRITICAL しきい値を設定します。設定する値については上限・下限しきい値が設定可能です。なお、しきい値につきましては取得した値全てに利用されます。
- active
  - この項目を false にしている場合は監視ステータスが「WARNING」または「CRITICAL」の場合のみ結果を送信します。True に設定している場合は監視ステータスに関わらず、監視結果を送信します。

7.1.3 X-MON 出力プラグインのインストール及び設定

X-MON 出力プラグインにつきましては RubyGems では配布しておりません。そのため、ファイルを直接設置してインストールする必要があります。
また、監視結果の送信に NSCA を使用しておりますので、X-MON 側で NSCA を受信できるように設定しておく必要があります。

7.1.3.1 X-MON 側の設定

X-MON 側で NSCA を受信し、監視結果を表示する設定を行います。

7.1.3.1.1 ポートの解放

NSCA を受信するためには NSCA がデータを受け取るポートを開放しておく必要があります。
X-MON がデフォルトで使用している NSCA のポート番号は「5667」となりますので、ファイアウォールなどで「5667」を開ける必要があります。

7.1.3.1.2 NSCA 受信設定

X-MON では NSCA を受信する際の設定を管理画面から行うことが出来ます。
設定画面については 管理者メニュー > その他設定 > 外部連携 > 分散監視(NSCA)受信設定 より進むことができます。

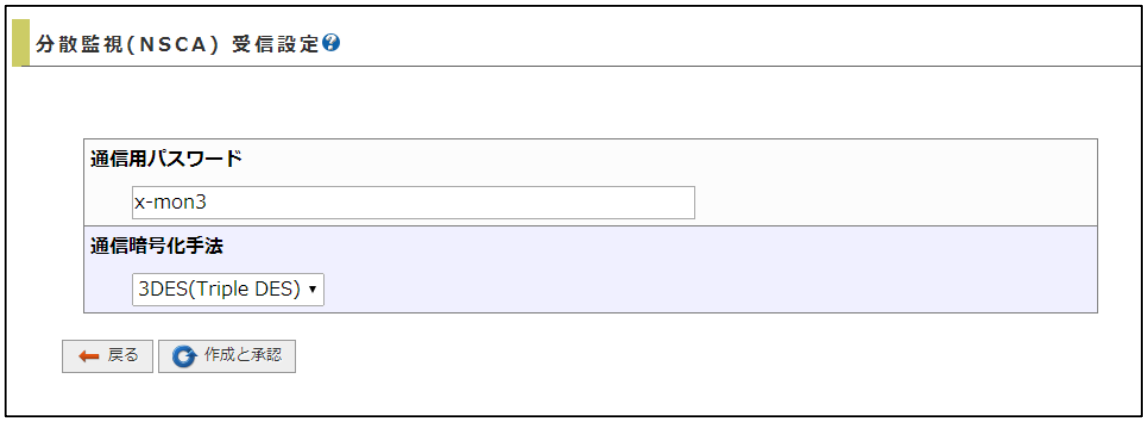


図 10 X-MON 分散監視（NSCA) 受信設定画面

この画面にある「通信用パスワード」及び「通信暗号化手法」をメモしておいてください。この項目については監視結果の送信側と受信側で合わせる必要があります。

7.1.3.1.3　　サービスの追加

X-MON 出力プラグインが結果を通知する監視サービスを作成する必要があります。サービスの設定については基本的に通常の設定と変わりませんが、以下の点を考慮して頂くようにお願いいたします。

- 基本設定
  - サービス監視用コマンド

    fluentd を用いたログ監視用の監視プラグインはございません。そのため、サービス監視用コマンドには「監視サポートコマンド」の「ダミープラグイン」をご利用ください。

図 11 X-MON サービスの作成 - 基本設定画面

- 監視設定
  - アクティブチェック：無効
  - パッシブチェック：有効
    - fluentd から送信された結果により監視を行いますので、定期的なチェックを行わないのでパッシブチェックにて監視を行うように設定します。
  - 試行回数：1 回
    - 試行回数は何回連続で障害ステータスを受信すると障害と確定するかを設定します。障害が確定するまでは通知やエスカレーション実行などの処理は行いません。ログ監視ではエラーログを複数回受信するということは通常ありません。よって、試行回数を 1 回にして 1 回でもログを受信すれば警告を上げるように設定することを推奨致します。

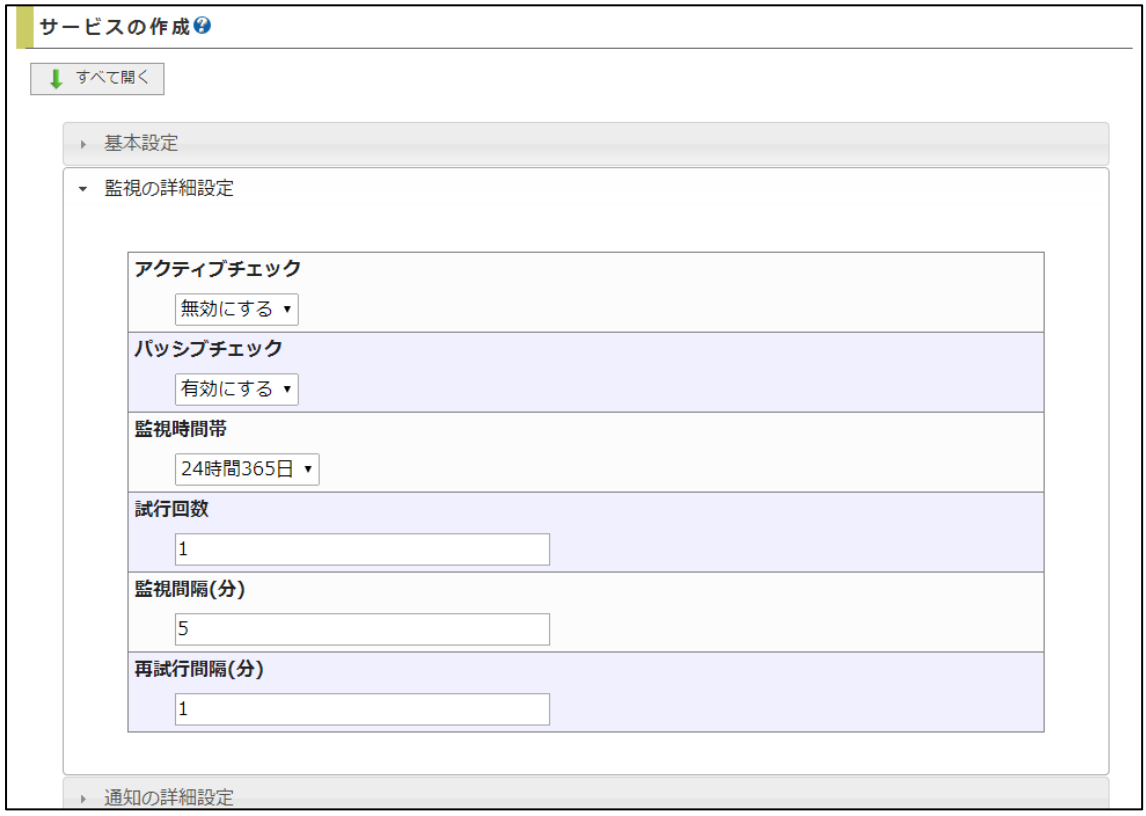


図 12 X-MON サービスの作成 - 監視の詳細設定画面

- 高度な設定
  - volatile サービス
    - ログ監視を行っている場合に、volatile サービスが無効になっている場合は障害発生中に新たに障害となるログを検知しても通知などを行いません。障害発生中でも新たに発生したログを検知したい場合につきましては、volatile サービスを有効にする必要があります。単に障害発生と障害復旧の通知のみを行いたい場合につきましては、有効にする必要はございません。
  - フレッシュネスチェック
  - フレッシュネスしきい値
    - fluentd では外部から監視結果を受けてから監視を行うため、監視結果を受け取らない場合には何も行いません。そのため、定期的にログ情報を取得するようなログ監視を行っている場合に、fluentd などに問題があり監視結果が送信されない状態になっても気付くことが難しくなります。
    - フレッシュネスチェックを使用すると、フレッシュネスしきい値で指定した時間を超えても監視結果が送信されてこない場合に強制的に監視プラグインに設定されている内容を実行いたします。そのため、この設定例ではフレッシュネスチェックが起動したタイミングで「ダミープラグイン」が実行され、ステータス及びステータス情報が監視プラグインで設定した内容に変更されます。

7.1.3.2 fluentd サーバ側の設定

fluentd サーバ側ではプラグインの設置と NSCA の送信設定を行う必要がございます。

7.1.3.2.1 X-MON 出力プラグインのダウンロード

X-MON 出力プラグインにつきましては、X-MON サポートサイトにて配布しております。それをダウンロードして頂き、X-MON へ結果を送信する fluentd サーバに送り展開してください。

7.1.3.2.2　X-MON 出力プラグインファイルの設置

展開したディレクトリには以下のようにファイルが存在します。

```
[ out_xmon ]　… 数値監視用プラグイン
[ out_xmon_log ] … 文字列監視用プラグイン
[x-mon]
 [ send_nsca ]　… X-MON に監視結果を送信する際に使用するコマンド
 [ send_nsca.cfg ]… send_nsca の設定ファイル
```

「out_xmon」と「out_xmon_log」ファイルについては「/etc/td-agent/plugin」に設置します。また、権限につきましては、fluentd を実行しているユーザが読み込みを行える権限を付与する必要があります。

「x-mon」ディレクトリについては「/etc/td-agent/」に設置します。

また、「send_nsca」については fluentd を実行しているユーザがコマンドを実行できる権限が必要となりますので、実行権限を付与してください。

7.1.3.2.3　send_nsca を使用する準備

send_nsca を利用するためには、「libmcrypt」パッケージが必要となります。

これは RHEL や CentOS の標準パッケージでは配布されておりません。

そのため、EPEL と呼ばれる外部のレポジトリよりパッケージをダウンロードするか、パッケージを直接ダウンロードしてサーバにパッケージをインストールする必要がございます。

7.1.3.2.4　send_nsca の設定ファイル修正

各環境に合わせて send_nsca の設定を変更する必要がございます。

先ほど設置した[send_nsca.cfg]の中にある「password」と「encryption_method」を先ほど設定した「通信用パスワード」と「通信暗号化手法」を設定します。

7.2 X-MON と fluentd の設定例

実際に X-MON と fluentd との連携を行うための設定例をご紹介します。

7.2.1　指定した IP アドレス以外から SSH の接続要求がないか監視をしたい

文字列監視用プラグインを使用することでログ情報から指定した IP アドレス以外からアクセスが無いかを監視することが出来ます。

この例では secure ログより SSH の接続要求があった接続元 IP アドレスを用いて監視を行います。

7.2.1.1 必要なプラグインのインストール

fluentd を用いたログ監視には文字列監視用プラグインを使用します。インストール方法については「7.1.3.2 X-MON 出力プラグインのダウンロード」をご参照ください。

7.2.1.2 Secure ログを fluentd に取り入れる設定

今回の例としては以下のログを使用します。

```
Mar 27 16:00:51 www1 sshd[28993]: Accepted password for root from 192.0.2.100 port 54503 ssh2
```

secure ログより fluentd にログ情報を取り入れる設定を行います。設定例は以下の通りです。

```
<source>
    type tail
    path /var/log/secure
    pos_file /var/log/td-agent/secure.pos
    tag www1.secure
    format /sshd.+[^:]:.+for (?<user>[^ ]+) from (?<ip>[^ ]+)/
</source>
```

実際に取り込んだログ情報は以下のようになっています。

```
{
 “user”: “root”,
  "ip": "192.0.2.100"
}
```

7.2.1.3 取り込んだログより監視を行う設定

今回の設定内容としては secure の全ログ情報より接続元の IP アドレスが「192.168.50.0-192.168.50.255」以外でかつユーザ名が「development」ではない接続に対して警告を上げるように設定を行います。

設定例は以下の通りです。

```
<match *.secure>
  type xmon_log
  xmon 192.168.100.51
  host WebServer
```

```
  service UNAUTHORIZED_ACCESS
  exclude1 ip 192¥.168¥.50¥.
  exclude2 user development
  status WARNING
  target user,ip
</match>
```

各設定内容の説明は以下の通りです。

```
exclude1 ip 192¥.168¥.50¥.
```

この設定では接続元 IP アドレスが「192.168.50.**」であるログ情報については除外する設定になります。

```
exclude2 user development
```

この設定ではユーザ名が「development」であるログ情報については除外する設定になります。

よって、この設定では接続元 IP アドレスが「192.168.50.**」以外であり、ユーザ名が「development」以外のユーザからの接続に対して警告を上げる設定となります。

7.2.1.4 再起動

設定が完了すれば、fluentd を再起動してください。再起動が完了すると、設定が反映されます。

7.2.1.5 表示例

以下が、「admin」ユーザで IP アドレス「192.168.0.24」のサーバより接続があった場合に X-MON 上で表示される内容です。「target」項目に「user」と「ip」を選択していますので、「user」と「ip」の両方がステータス情報に表示されます。

また、「status」項目を「WARNING」にしているためステータスは WARNING になります。

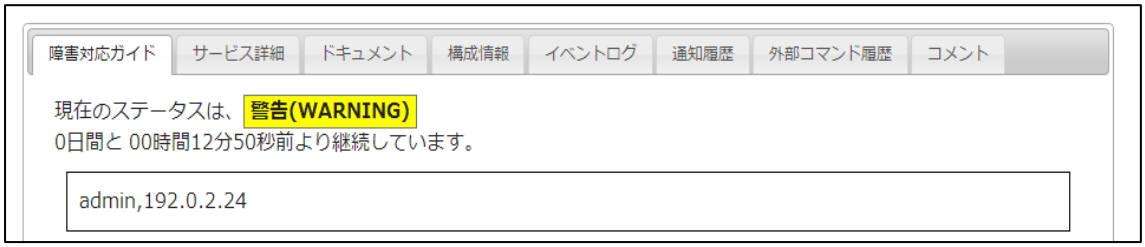


図 13 X-MON UNAUTHED_ACCESS サービス情報画面

7.2.2 MySQL のスロークエリを監視したい

「mysqlslowquery」プラグインを使用することで MySQL のスロークエリログをログ情報として取得することができます。このプラグインを用いて MySQL のスロークエリ監視を行います。

なお、監視を行う上で以下の設定がされていることを前提とします。

- MySQL がインストールされている。
- MySQL がスロークエリログを出力するように設定されている。
- MySQL を実行しているサーバに fluentd がインストールされている。

7.2.2.1 必要なプラグインのインストール

この監視では「mysqlslowquery」プラグインを使用します。「mysqlslowquery」プラグインについては RubyGems で配布されていますので、以下のコマンドでインストールします。

```
# /usr/lib64/fluent/ruby/bin/gem install fluent-plugin-mysqlslowquery
```

7.2.2.2 MySQL のスロークエリログを fluentd に取り入れる設定

インストールした「mysqlslowquery」プラグインを使用してスロークエリログを fluentd に取り入れる設定を行います。設定例は以下の通りです。

```
  <source>
  type mysql_slow_query
  path /path/to/mysqld-slow.log
  tag mysqld.slow_query
</source>
```

Path の項目についてはスロークエリログが出力されているファイルのパスを指定します。
実際に取り込んだログ情報は以下のようになっています。

```
{
  "user": "root[root]",
  "host": "localhost",
  "host_ip": "",
  "query_time": 0.000270,
  "lock_time": 0.000097,
```

```
  "rows_sent": 1,
  "rows_examined": 0,
  "sql": "SET timestamp=1317619058; SELECT * FROM life;"
}
```

7.2.2.3 取り込んだログ情報より監視を行う

取得したログ情報を基に数値監視用プラグインを用いて監視を行います。

設定は以下の通りです。

対象サービス：ホスト【MySQLServer】のサービス【SLOW_QUERY】

```
<match mysqld.slow_query>
  type xmon
  xmon 192.168.10.51
  host MySQLSever
  service SLOW_QUERY
  description_name_type fixed
  description_name query_time
  keys query_time
  warning 1
  critical 2
</match>
```

この設定を行うことで、ログ情報より「query_time」の値を用いて監視を行い、warning または critical で設定した値を上回った際にアラートが発生するようになります。また、グラフについては「query_time」の値を用いて描写されます。

7.2.2.4 再起動

設定が完了すれば、fluentd を再起動してください。再起動が完了すると、設定が反映されます。

7.2.2.5 類似な利用例

類似した利用例と致しましては CloudWatch から取得してきた値を用いた監視がございます。CloudWatch から値を取得する方法につきましては「6.2.8 CloudWatch の値を取得したい」をご参照ください。

7.2.3 ステータスコード毎のアクセス数を集計したい。

「datacounter」プラグインを使用することで指定した要素の値が正規表現にマッチしたログ情報の数をカウントアップすることができます。この機能を用いて Apache のアクセスログよりステータスコード毎のアクセス数を監視します。

7.2.3.1 必要なプラグインのインストール

この設定で使用する「datacounter」プラグインをインストールします。「datacounter」プラグインは RubyGems でインストールされていますので、以下のコマンドを使用してインストールすることが出来ます。

```
# /usr/lib64/fluent/ruby/bin/gem install fluent-plugin-datacounter
```

7.2.3.2 Apache のアクセスログを fluentd に取り入れる設定

先述の設定例に記載した「apache のアクセスログを fluentd サーバに取り込む」の項目をご参照ください。

実際に取り込んだログ情報は以下のようになっています。

```
{
  "host":"192.168.13.72",
  "user":null,
  "method":"GET",
  "path":"/",
  "code":200,
  "size":14447,
  "referer":null,
  "agent":"Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/33.0.1750.146 Safari/537.36"
}
```

7.2.3.3 ステータスコード別のアクセス数を集計する設定

インストールした「datacounter」プラグインを使用してステータスコード別のアクセス数を集計します。実際の設定例は以下の通りです。

```
<match *.apache.access>
  type datacounter
  count_interval 60
  count_key code
  tag www.access.status
  pattern1 2xx ^2¥d¥d$
  pattern2 3xx ^3¥d¥d$
  pattern3 4xx ^4¥d¥d$
  pattern4 5xx ^5¥d¥d$
</match>
```

各設定項目は以下の通りです。

- Count_interval
  - 集計を行う周期を設定します。単位は秒です。
- Count_key (必須)
  - 集計を行う際に条件を指定する要素を指定します。
- Pattern[1-20]
  - 集計を行う条件を指定します。指定方法は半角スペース区切りで「patternN 識別子 検索条件」となります。検索条件については正規表現を利用することができます。
- Tag, tag_prefix, input_tag_remove_prefix
  - 出力するログ情報に付与するタグ名を設定します。
    tag を設定した場合はタグ名が指定した文字列に置き換わります。
    tag_prefix を指定した場合は入力されたログ情報のタグ名の先頭に指定した文字列が付与されます。
    input_tag_remove_prefix を指定した場合は指定した文字列をタグ名から除去します。

出力されるログ情報は以下となります。

```
2014-03-16T17:18:14+09:00 x-mon.access_status
{
  "unmatched_count":0,
  "unmatched_rate":0.0,
  "unmatched_percentage":0.0,
  "2xx_count":3,
```

```
  "2xx_rate":0.05,
  "2xx_percentage":4.615384615384615,
  "3xx_count":62,
  "3xx_rate":1.03,
  "3xx_percentage":95.38461538461539,
  "4xx_count":0,
  "4xx_rate":0.0,
  "4xx_percentage":0.0,
  "5xx_count":0,
  "5xx_rate":0.0,
  "5xx_percentage":0.0
}
```

7.2.3.4 集計した結果を X-MON に送信する設定

集計した結果より監視結果を X-MON に送信します。実際の設定例は以下の通りです。

例での X-MON の環境は以下の通りです。

X-MON サーバ：192.168.100.52

対象サービス：「WebServer」ホストの「STATUS_CODE」サービス

```
  <match www.access.status>
  type xmon
  xmon 192.168.100.52
  host WebServer
  service STATUS_CODE
  description_name_type fixed
  description_name 2xx,3xx,4xx,5xx
  keys 2xx_count,3xx_count,4xx_count,5xx_count
</match>
```

この設定を行うことで、「datacounter」プラグインより出力された情報の中から keys で指定した値を用います。この設定ではしきい値を設定していませんので、監視は行わずグラフの生成のみを行います。

7.2.3.5 再起動

設定が完了すれば、fluentd を再起動してください。再起動が完了すると、設定が反映されます。

7.2.3.6 表示例

以下が実際に監視をおこなった際の画面表示です。ステータス情報には直近で送られてきた情報が表示されます。

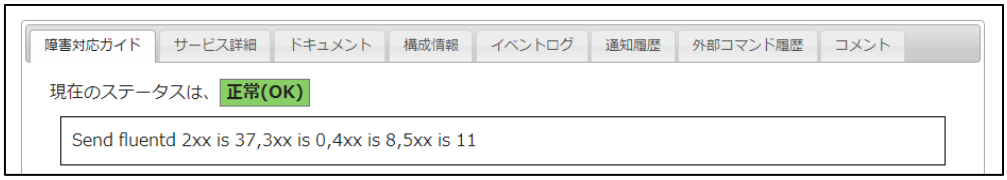


図 14 X-MON STATUS_CODE サービス情報画面

7.2.4 送信元 IP アドレス毎のアクセス数を監視したい。

「uniq_count」プラグインを使用することで指定した要素の値毎にログの数を集計することができます。今回の例ではアクセスログの送信元 IP アドレス毎にログの数を集計します。

7.2.4.1 必要なプラグインのインストール

この監視で使用する「uniq_count」プラグインについては RubyGems で配布されておりません。そのため、プラグインのインストールにつきましては「6.2.1.2 直接ファイルを設置する方法」をご参照ください。対象のプラグインを配布しているサイトは以下になります。

https://github.com/KazkiMatz/fluent-plugin-uniqcount

7.2.4.2 Apache のアクセスログを fluentd に取り入れる設定

先述の設定例に記載した「apache のアクセスログを fluentd サーバに取り込む」の項目をご参照ください。

今回は必要な情報を取得するために設定項目の「format」を以下のように変更します。

```
format /(?<host>[^ ]+) [^ ]+ [^ ]+ ¥[(?<date>[^ ]+) [^¥]]+¥] ¥"(?<method>[^ ]+)
(?<path>[^ ]+) [^ ]+ (?<code>[^ ]+)/
（改行されて見えていますが、実際は 1 行です）
```

実際に取り込んだログ情報は以下のようになっています。

```
{
  "host":"192.168.13.72",
  "date":27/Mar/2014:14:14:38,
```

```
  "method":"GET",
  "path":"/",
  "code":200,
}
```

7.2.4.3 接続元 IP アドレス毎にログを集計する設定

インストールした「uniq_count」プラグインを使用して接続 IP アドレス毎にログを集計します。
設定例は以下の通りです。

```
<match *.apache.access>
  type uniq_count
  list1_label peakAccess
  list1_key1 host
  list1_key2 date
  list1_span 60
  list1_out_tag x-mon.access_count
  list1_out_num 100
 list1_out_interval 60
</match>
```

各設定項目は以下の通りです。

- listN_label (必須)
  - 集計した結果に付与するラベル名を指定します。
- listN_key1 (必須)
  - グルーピングを行う要素名を指定します。
  - 今回の場合は接続元 IP アドレスでグルーピングを行うので「host」要素を指定しています。
- listN_key2 (必須)
  - グルーピングしたログの中から指定した要素の異なる値でカウントをします。複数ログが存在する場合でも、ここで指定した値が同一であれば同一のログとしてみなされます。
- listN_span
  - 集計を行う間隔を指定します。デフォルトでは 60 秒間のログを集計の対象としています。デフォルトは 60 秒です。

- listN_out_tag (必須)
  - 出力した際のログ情報に付与するタグ名を指定します。
- listN_out_num
  - グルーピングしたログの最大出力数を設定します。デフォルトは 10 です。
- listN_out_interval
  - 集計結果の出力間隔を指定します。デフォルトは 1 秒です。

「listN」の N には数字が入ります。N を違う数字に変えることで同様のログ情報から異なる方法で集計を行うことが出来ます。用途と致しましては同じログ情報から日別や週別でそれぞれ集計結果を出す際に利用します。
また、それぞれの結果は異なるレコードで出力されます。

実際に出力されるログは以下の通りです。

```
{
 "label":"peakAccess",
 "ranks":
     [
      {
       "key1":"192.168.198.3",
       "rank":0,
       "key2_count":469,
       "key2_uniq_count":60
      },
      {
       "key1":"192.168.198.9",
       "rank":1,
       "key2_count":458,
       "key2_uniq_count":60
      },
      {
       "key1":"192.168.198.0",
       "rank":2,
       "key2_count":442,
```

```
        "key2_uniq_count":60
       },
       {
        "key1":"192.168.198.1",
        "rank":3,
        "key2_count":428,
        "key2_uniq_count":60
       }
      ],
 "at":1393945833
}
```

7.2.4.4 集計した結果を X-MON に送信する設定

集計した結果より監視結果を X-MON に送信します。実際の設定例は以下の通りです。

例での X-MON の環境は以下の通りです。

X-MON サーバ：192.168.100.52

対象サービス：「WebServer」ホストの「PEAK_ACCESS」サービス

```
<match www.access.peak>
  type xmon
  xmon 192.168.100.52
  host WebServer
  service PEAK_ACCESS
  description_name_type auto
  description_name ranks.*.key1
  keys ranks.*.key2_count
  warning 500
  critical 600
  active false
</match>
```

この設定を行うことで、「uniq_count」プラグインより出力された情報の中から ranks 要素に含まれる全レコードの「key2_count」の値を用いて監視を行います。また、各値の識別子としては同

様に ranks 要素に含まれる全レコードの「key1」の値を識別子として利用します。
また、「active」要素を false に設定していますので、この監視は監視ステータスが WARNING または CRITICAL となった場合のみ監視結果を送信します。
なお、「key2_count」と「key2_uniq_count」の違いにつきましては、「key2_count」については「key1」で指定した要素でグルーピングしたログの総数となり、「key2_uniq_count」については「key2」で指定した要素でユニークな値を集計した結果になります。

7.2.4.5 再起動

設定が完了すれば、fluentd を再起動してください。再起動が完了すると、設定が反映されます。

7.2.4.6 表示例

以下が X-MON に実際通知される例です。この例では IP アドレス「192.0.20.47」のホストより 1 分間に 625 回のアクセスがあったことを示します。
なお、この設定では障害を検知すると手動で復旧させるまで CRITICAL の状態が続きます。「パッシブチェックの結果を送信」を用いて監視を復旧させてください。

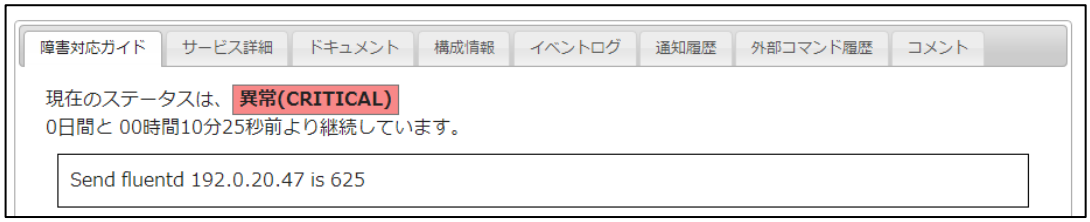


図 15 X-MON PEAK_ACCESS サービス情報画面

7.2.5 X-MON のログを fluentd に転送したい

X-MON のログを fluentd に転送することで他のログと同様に集約や蓄積を行うことが出来ます。また、本設定を行う場合につきましては X-MON サーバに fluentd をインストールする必要があります。

7.2.5.1 必要なプラグインのインストール

この設定で使用するプラグインにつきましては、fluentd に標準で付属しているものですので別途インストール作業を行う必要はございません。

7.2.5.2 X-MON のログを fluentd に取り入れる設定

X-MON のログを fluentd に取り入れるには「in_tail」プラグインを使用します。
「in_tail」プラグインの設定項目などにつきましては「6.2.6.1 アクセスログファイルからログ情報を fluentd に取り入れる設定」をご参照ください。

X-MON ログよりサービスアラートのログを取り入れる際の設定は以下になります。

```
<source>
    type tail
    path /var/log/x-mon/x-mon.log
    pos_file /var/log/td-agent/x-mon.log.pos
    tag x-mon.alert.service
    format /^¥[(?<date>[^*¥]]+)¥] SERVICE ALERT:
  (?<host>[^;]+);(?<service>[^;]+);(?<state>[^;]+);(?<statetype>[^;]
  +);(?<attempt>[^;]+);(?<msg>[^*]+)/
</source>
```

（format については改行されて見えていますが、実際は 1 行です）

Format については取得したいログ情報に合わせて変更する必要があります。

実際に取り込んだログ情報は以下のようになっています。

```
2014-03-12T13:20:10+09:00   x-mon.serviceAlert
{
  "date":"1394598010",
  "host":"WebServer",
  "module":"PING",
  "state":"CRITICAL",
  "statetype":"HARD",
  "attempt":"3",
  "msg":"PING CRITICAL - Packet loss =100%, RTA = 0.52 ms"
}
```

7.2.5.3 取り込んだログを転送する設定

この設定につきましては「out_forward」プラグインを使用した転送になります。

設定方法につきましては「6.2.6.2 取り込んだログ情報を fluentd サーバへ転送する設定」をご参照ください。