

2018年6月12日  
株式会社インプレスR&D  
<https://nextpublishing.jp/>

Web サービス監視を効率化！  
『Mackerel ではじめるお手軽 Web サービス監視』発行  
技術書典シリーズ、6月の新刊

インプレスグループで電子出版事業を手がける株式会社インプレス R&D は、『Mackerel ではじめるお手軽 Web サービス監視』(著者:大中 浩行)を発行いたします。

『Mackerelではじめるお手軽Webサービス監視』  
<https://nextpublishing.jp/isbn/9784844398301>



著者: 大中 浩行  
小売希望価格: 電子書籍版 1400 円(税別) / 印刷書籍版 1600 円(税別)  
電子書籍版フォーマット: EPUB3 / Kindle Format8  
印刷書籍版仕様: B5 判 / カラー / 本文 90 ページ  
ISBN: 978-4-8443-9830-1  
発行: インプレス R&D

<< 発行主旨・内容紹介 >>

【Mackerel でお手軽 Web サービス監視！】

本書は株式会社はてなが提供するサーバー監視サービス「Mackerel」について、導入方法やその特徴、現場で活用するノウハウを解説した書籍です。Web サービスの監視に必要な情報をコンパクトにまとめています。

(本書は、次世代出版メソッド「NextPublishing」を使用し、出版されています。)

# Mackerel の導入方法とセットアップを詳しく紹介

図3.2: オルガニゼーションの入力



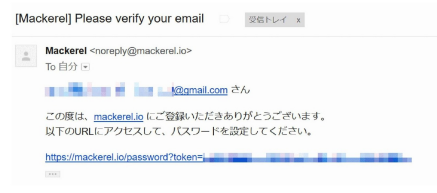
続いてプランの選択に移ります。オルガニゼーションの登録から2週間はTrialプランとして全ての機能を制限なく使用することができますので、Trialプランを選択します。(図3.3)

図3.3: プランの選択



プランの選択が完了すると、登録したメールアドレスに [Mackerel] Please verify your email という Subject のメールが届きます。メール文中の案内にしたがってアカウントのパスワードの設定を行うと、初期登録は完了です。図3.4

図3.4: パスワードの設定



### 3.2 mackerel-agent のセットアップ

mackerel にログインすると、ダッシュボードの左下に、「スタートガイド」「新規ホストの登録」というリンクがあります。セットアップは、このリンクから出るガイダンスに従って行います。(図3.5)

#### 機能のURL

このリンクのURLは `https://mackerel.io/orgs/<オルガニゼーション名>/instruction-agent` となりオルガニゼーションごとに変わります。Mackerel では `https://mackerel.io/my/instruction-agent` というURLにアクセスすることで、ログインセッションが最後に参照していたオルガニゼーションにリダイレクトされます。本書では、機能のURLを表記する際この形式で表記します。

#### mackerel-agent のサポート環境

Mackerel で監視を行うためのエージェントである mackerel-agent は以下の環境をサポート

# 様々なプラグインを紹介し、サーバーの監視方法を解説

## 第6章 チェック監視

ここまで紹介してきた同様接続数やサーバーのさまざまなリソースの状況など、サーバー上のメトリックを時系列によって監視する方法をメトリック監視と呼びます。これに対し、サーバーのヘルスチェックの状況やログファイル内のエラーログ出力への監視など、サーバー上の時系列によらない個別のステータスをチェックした結果の監視を、Mackerel ではチェック監視と呼んでいます。

### 6.1 メトリック監視とチェック監視

以前は、監視と言えばチェック監視が中心でした。それは、メトリックを時系列で取得するには、ディスク領域や監視に用いるサーバーの処理性能などの制約があったことも理由の一つです。

しかし、チェック監視の運用が生まれる背景には、汎用機のバッチ処理でエラーが発生したら運用者が対応する、ハードウェアの診断機能が故障を検出したらベンダーに保守対応を依頼する、といったWebサービスの普及する前の運用フローがあります。そのようなフローをそのままWebシステムに適用した結果、チェック監視中心の運用フローが生まれたのです。

ですが、ITサービスの社会への浸透と、仮想化技術やパブリッククラウドの普及の中で、継続的に成長を続けるITサービスの運用を続けるうえで、アクセス負荷の変化によるスケールアウト・スケールインなどインフラ構成を能動的に変動できるようにすることへのニーズが高まってきました。

その中で、ホスト上やサービスに関する性能や可用性に関するメトリック収集と監視を継続的に、可視化することが必要だという考え方が普及してきました。

また、時系列データベースなどの技術の発展やハードウェア性能の向上などにより、以前は難しかったメトリックの監視を中心に据えた監視システムの運用が可能になってきました。

このため、Mackerel はメトリック監視を中心に据えた設計になっています。

しかしエラーログの監視や外形監視などのサービスのエンドポイントやヘルスチェックの監視など、メトリックによらない監視へのニーズも依然として存在しています。この場合はチェック監視の機能を用いた監視を行います。

### 6.2 go-check-plugins

Mackerel では、チェック監視に対応した公式プラグイン集がオープンソース (Apache License

2.0)<sup>1)</sup>で提供されています。インストールは、yumコマンドないし apt コマンドで行います。(リスト6.1)

リスト 6.1: プラグインのインストール

```
sudo yum install mackerel-check-plugins
```

各プラグインのコマンドは /usr/bin/以下にインストールされます。go-check-plugins では、次のプラグインが提供されています。

表6.1: go-check-plugins のプラグイン一覧

プラグイン名称	チェック対象
check-amazon-sqs-queue-size	Awsam Simple Queue Service (SQS) のキューサイズ
check-cert-file	SSL 等の証明書運用
check-disk	ファイルシステムのパーテーション等の監視
check-elasticsearch	Elasticsearch の /_cat/master/health の視覚情報
check-file-age	ファイルのタイムスタンプならびにサイズ
check-file-size	指定したディレクトリ内のファイルサイズ
check-http	http 接続
check-jmx-jolokia	JMX のリモートアクセスツール Jolokia 経由で取得できるJVMの指標
check-ldap	LDAP のレスポンス時間
check-load	ロードアバレッジ
check-log	行単位で出力されるログファイル
check-mail	メールサーバー (MTA) の監視キュー
check-masterha	MHA (Master HA) for MySQL
check-memcached	memcached
check-mysql	MySQL
check-nfs	NTP のオフセット
check-nfs-service	Windows サービス (以前 NT サービスと呼ばれていたもの)
check-postgresql	PostgreSQL
check-process	プロセスの死活監視等
check-redis	Redis
check-smb	Apache Smb
check-ssh	SSH のヘルスチェック
check-step	TCP/UNIX ドメインソケットの監視
check-uptime	プロセスの起動時間
check-windows-eventlog	Windows イベントログ

<sup>1)</sup> <https://github.com/mackerelio/go-check-plugins>  
2) go-check-plugins の詳細は、Mackerel のサイト上にある「OSS」の「プラグイン集の提供形態」の「Mackerel のプラグイン集」のページを参照してください。

# バッチ処理において運用に必要な仕組みを Mackerel で作る方法を紹介

## 第9章 バッチ処理のロギングと監視

Linux サーバーにおいて、未だに cron によるバッチ処理はシステム構築で重要な地位を占めています。例えばバックアップ処理や、ログファイルのアーカイブ・ローテーションなどです。バッチ処理は、エンドユーザーに対してユーザーインターフェースを持ちません。このため、実行の結果を確認するために、実行結果やログに対する監視は重要な役割があります。しかし、モダンなプログラミング言語やフレームワーク上でのシステム開発と比べて、シェルスクリプトによるシステムの監視・運用は難易度が高いものとなっています。それは、例外処理やロギングフレームワーク、エラー発生時のスタックトレース出力など、モダンなフレームワークが有している機能をシェルスクリプトが有していないためです。このような運用に必要な仕組みをフルスタックで作り込むことは、構築と運用に関するコストのバランスが悪いバッチ処理を生み出します。本章では、シェルスクリプトによるバッチ処理において、OS が提供する枠組みの中で運用に必要な仕組みを作り込み、Mackerel での監視に組み込む方法について述べます。なお、本章では、GNU Bash のバージョン 4.2.46(2)-release(x86\_64-redhat-linux-gnu) を動作環境とし、bash のインタプリタが /bin/bash であるものとして記述を行っています。

### 9.1 バッチ処理の監視に必要な要素

#### 例外処理

特にオプションを付与していないシェルスクリプトには、個々のコマンド実行ごとにエラーハンドリングが必要です。しかし、実装時の負担が大きいため、`-e` および `-u` オプションで、コマンドのエラー終了や未定義変数の参照などの実行時エラーを補足できるようにします。この際呼び出し先のメインのシェルスクリプトに `-e` オプションを付与すると、エラー発生時にスクリプトの実行が打ち切られてしまうため、処理全体に対する結果のハンドリングができなくなります。メイン処理となるシェルスクリプトとそこから呼び出される処理を実装するシェルスクリプトを分割し、処理を実装するシェルスクリプトに `-e` オプションを付与することで、結果のハンドリングができるようになります。

#### ロギング

ログについては、`logger` コマンドを使用して `syslog` に集約します。前述のスクリプトでは、ログは全て標準出力に `echo` コマンドで出力したものを `logger` コマンドで出力することで、スクリプトのログ出力の実装を単純にします。

#### エラー発生時のスタックトレース

シェルスクリプトにはスタックトレースという機能が存在しませんが、`-x` オプションの出力を標準エラー出力で補足してエラー発生時にログ出力することで、処理をトレース可能にします。

#### 監視

監視は、`syslog` の出力を Mackerel でチェック監視することで行います。

### 9.2 実装例

以下に示す処理は、Jira の出力するアプリケーションログファイルについて、5 世代以上前のファイルは圧縮するバッチ処理です。

#### メイン処理

リスト 9.1 のシェルスクリプトがメイン処理の実装です。処理内で標準出力をログ出力として、標準エラー出力を `-x` オプションによるトレースの出力として取得します。スクリプトの終了コードによって結果を判定し、ログの出力を行います。`logger` コマンドの `-t` オプションに `$(0)` による実行シェルスクリプト名を埋め込むことで、ログ出力内でスクリプトの区別がつかないようにしています。

リスト 9.1: メイン処理 (jira-log-compress-atlassian-jira.sh)

```
#!/bin/bash

set -u

# 一時ファイルの作成
output=$(mktemp)
error=$(mktemp)

# 処理の呼び出し
/usr/local/bin/jira-log-compress.sh \
  "atlassian-jira.log" 1>${output} 2>${error}
RET=$?
cat $output |logger -is -t $(basename $0)

# 結果のログ出力
if [ $RET -eq 0 ]; then
  logger -is -t $(basename $0) "successful end."
```

## << 目次 >>

### 第1章 なぜゆえの監視

- 1.1 誰のためのモニタリング
- 1.2 監視にもとめられるもの

### 第2章 Mackerel について

- 2.1 Mackerel の特徴① 導入の容易さ
- 2.2 Mackerel の特徴② プラグイン開発の容易さ
- 2.3 Mackerel の特徴③ 課金体系
- 2.4 用語について

### 第3章 Mackerel の導入

- 3.1 Mackerel に登録する
- 3.2 mackerel-agent のセットアップ
- 3.3 Windows への mackerel-agent の導入
- 3.4 ユーザーの招待

### 第4章 Mackerel による Web システム監視

- 4.1 システムメトリック
- 4.2 公式プラグインのセットアップ
- 4.3 TOML
- 4.4 mackerel-agent.conf の設定の反映
- 4.5 mackerel-plugin-jvm による JVM の監視
- 4.6 プラグインをコマンドとして実行する
- 4.7 mackerel-plugin-inode による inode の監視
- 4.8 mackerel-plugin-postgres による PostgreSQL の監視
- 4.9 マルチコアの CPU とロードアベレージ

- 4.10 mackerel-plugin-multicore によるマルチコアの CPU 使用率監視
- 4.11 監視ルールの設定
- 4.12 監視の閾値
- 4.13 通知の設定
- 第 5 章 プラグイン開発による対象メトリックの追加
  - 5.1 プラグインによるメトリック監視
  - 5.2 プラグイン実行時の環境変数の設定
- 第 6 章 チェック監視
  - 6.1 メトリック監視とチェック監視
  - 6.2 go-check-plugins
  - 6.3 check-http プラグインによる死活監視
  - 6.4 check-log プラグインによるログ監視
  - 6.5 プラグイン開発によるチェック監視
  - 6.6 JSON 形式でのログ出力によるチェック監視の設定
  - 6.7 チェック監視の設定へのメモ記述
- 第 7 章 アクセスログの可視化によるサービス水準の監視
  - 7.1 mackerel-plugin-accesslog
  - 7.2 mackerel-plugin-accesslog によるアクセスログの集計
  - 7.3 サーバーダウンの場合のメトリック
- 第 8 章 URL 外形監視による Web サイトのヘルスチェック
  - 8.1 URL 外形監視の設定
  - 8.2 URL 外形監視のアラート通知
  - 8.3 User-Agent の設定
  - 8.4 IP アドレス帯域の設定
  - 8.5 Basic 認証の設定
- 第 9 章 バッチ処理のロギングと監視
  - 9.1 バッチ処理の監視に必要な要素
  - 9.2 実装例
- 第 10 章 API を使用したサービスメトリックの投稿
  - 10.1 サービスメトリックについて
  - 10.2 Azure Functions によるサービスメトリック作成
  - 10.3 グラフの共有
  - 10.4 サービスアノテーション
- 第 11 章 コマンドラインツール mkr
  - 11.1 API キーの設定
  - 11.2 mkr の機能
  - 11.3 ホストのステータスの更新
  - 11.4 メトリックの取得
- 第 12 章 まとめ

## << 著者紹介 >>

大中 浩行

TDD(テスト駆動開発)のワークショップ TDD Boot Camp(TDDBC)の主催など、TDDを始めとするアジャイル開発のプラクティスの普及活動に従事。本職はシステムインテグレーターでの、テスト自動化や継続的デリバリーに関する技術支援およびインフラ構築・運用。情報処理安全確保支援士(登録番号 第 004514 号) <http://hiroyuki.fieldnotes.jp/>

## <<販売ストア>>

### 電子書籍:

Amazon Kindle ストア、楽天 kobo イーブックストア、Apple iBookstore、紀伊國屋書店 Kinoppy、Google Play Store、honto 電子書籍ストア、Sony Reader Store、BookLive!、BOOK☆WALKER

### 印刷書籍:

Amazon.co.jp、三省堂書店オンデマンド、honto ネットストア、楽天ブックス

※ 各ストアでの販売は準備が整いしだい開始されます。

※ 全国の一般書店からもご注文いただけます。

### 【株式会社インプレス R&D】 <https://nextpublishing.jp/>

株式会社インプレス R&D（本社：東京都千代田区、代表取締役社長：井芹昌信）は、デジタルファーストの次世代型電子出版プラットフォーム「NextPublishing」を運営する企業です。また自らも、NextPublishing を使った「インターネット白書」の出版など IT 関連メディア事業を展開しています。

※NextPublishing は、インプレス R&D が開発した電子出版プラットフォーム(またはメソッド)の名称です。電子書籍と印刷書籍の同時制作、プリント・オンデマンド(POD)による品切れ解消などの伝統的出版の課題を解決しています。これにより、伝統的出版では経済的に困難な多品種少部数の出版を可能にし、優秀な個人や組織が持つ多様な知の流通を目指しています。

### 【インプレスグループ】 <https://www.impressholdings.com/>

株式会社インプレスホールディングス(本社：東京都千代田区、代表取締役：唐島夏生、証券コード：東証1部9479)を  
持株会社とするメディアグループ。「IT」「音楽」「デザイン」「山岳・自然」「モバイルサービス」を主要テーマに専門性の高いコンテンツ+サービスを提供するメディア事業を展開しています。

### 【お問い合わせ先】

株式会社インプレス R&D NextPublishing センター

TEL 03-6837-4820

電子メール: np-info@impress.co.jp