

AWS Dev Day Tokyo 延長戦

 **terraport**

株式会社テラポート

# 自己紹介



 terraport

株式会社テラポート

名前	山野 寛和
所属 / 役職	株式会社テラポート / CTO
Twitter	@refirio
GitHub	refirio

## 事業紹介

An isometric illustration of a university campus with various digital devices and people. A laptop in the foreground displays the text 'university applications online'. Other elements include a smartphone, a tablet, and a person sitting at a desk. The word 'Campus' is written in large, bold, black letters in the center.

# Campus

大学・専門学校向け Web出願システム

An isometric illustration of a school environment with digital devices and people. A laptop displays 'Can'dy LOCK', a tablet shows 'Can'dy APP', and another device shows 'Can'dy RESERVE'. A person is standing next to a laptop with 'Can'dy ENTRY' on the screen. The word 'Can'dy' is written in large, bold, black letters in the center.

# Can'dy

中学校・高等学校向け 生徒募集支援

An illustration showing three children looking at a large screen displaying a school building. The screen shows a clock and several blue water bottles. The children are standing in front of the screen. The text 'キャンディ・キッズ' is written in large, bold, black letters in the center.

# キャンディ・キッズ

子ども向け アクティブ・ラーニング

# その他 受託開発

プログラム作成・サーバ保守など

# 概要

## 課題

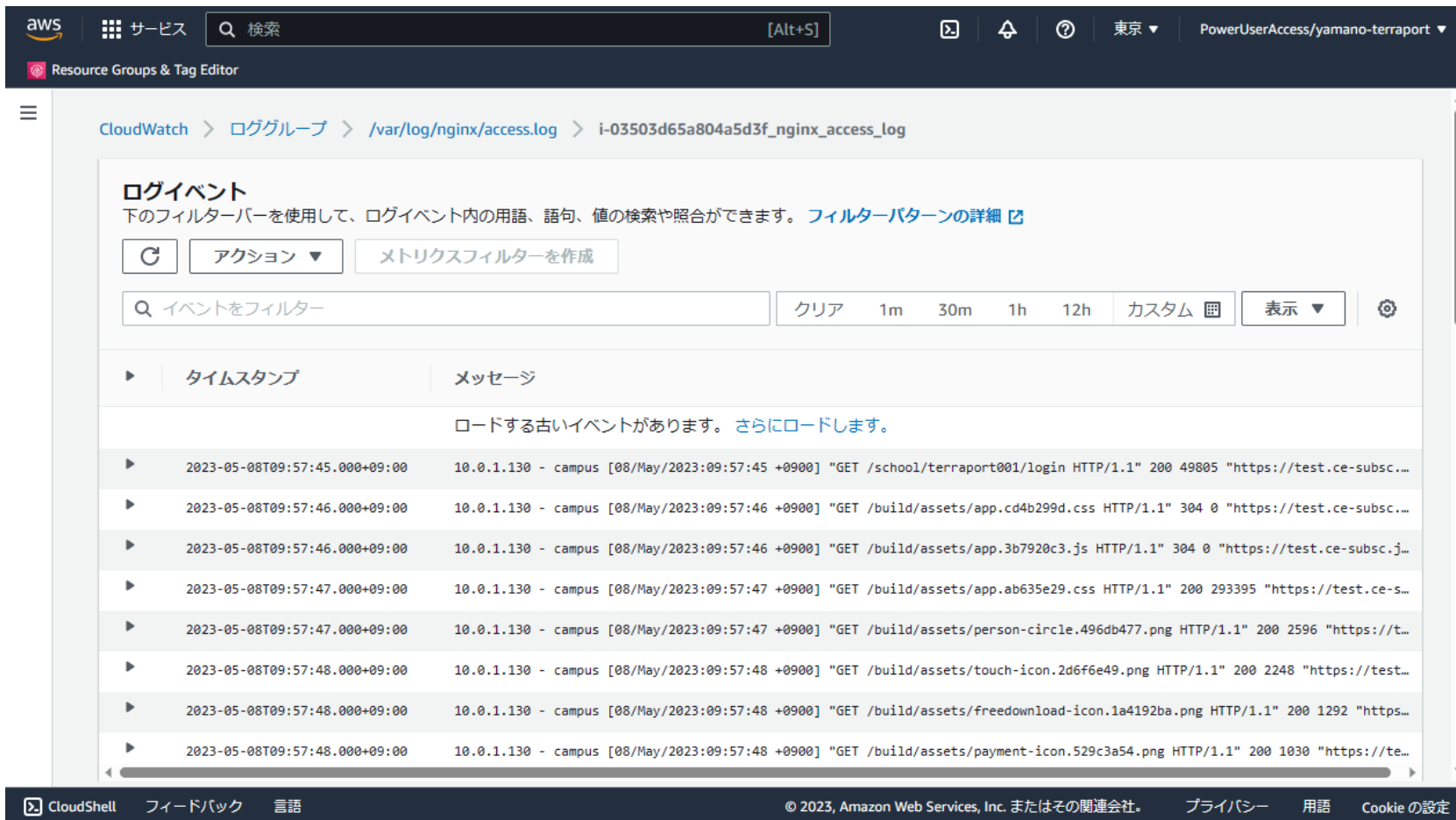
複数台構成のサーバで運用する際、サーバのログを手軽に確認したい。



## どうするか

- Amazon Elastic Compute Cloud (Amazon EC2) にSSHで接続して確認 ▲
- Amazon CloudWatch Logs に転送して確認 ●

# ただし、ログの閲覧画面は少々敷居が高い



The screenshot displays the AWS CloudWatch console interface for a log group. The breadcrumb navigation shows the path: CloudWatch > ロググループ > /var/log/nginx/access.log > i-03503d65a804a5d3f\_nginx\_access\_log. The main content area is titled "ログイベント" (Log Events) and includes a search bar with the text "イベントをフィルター" (Filter events) and a "クリア" (Clear) button. Below the search bar, there are time range filters: 1m, 30m, 1h, 12h, and カスタム (Custom). A "表示" (Display) button is also present. The log events are displayed in a table with two columns: "タイムスタンプ" (Timestamp) and "メッセージ" (Message). The messages are HTTP access logs from 10.0.1.130.

タイムスタンプ	メッセージ
2023-05-08T09:57:45.000+09:00	10.0.1.130 - campus [08/May/2023:09:57:45 +0900] "GET /school/terraport001/login HTTP/1.1" 200 49805 "https://test.ce-subsc...
2023-05-08T09:57:46.000+09:00	10.0.1.130 - campus [08/May/2023:09:57:46 +0900] "GET /build/assets/app.cd4b299d.css HTTP/1.1" 304 0 "https://test.ce-subsc...
2023-05-08T09:57:46.000+09:00	10.0.1.130 - campus [08/May/2023:09:57:46 +0900] "GET /build/assets/app.3b7920c3.js HTTP/1.1" 304 0 "https://test.ce-subsc.j..
2023-05-08T09:57:47.000+09:00	10.0.1.130 - campus [08/May/2023:09:57:47 +0900] "GET /build/assets/app.ab635e29.css HTTP/1.1" 200 293395 "https://test.ce-s...
2023-05-08T09:57:47.000+09:00	10.0.1.130 - campus [08/May/2023:09:57:47 +0900] "GET /build/assets/person-circle.496db477.png HTTP/1.1" 200 2596 "https://t...
2023-05-08T09:57:48.000+09:00	10.0.1.130 - campus [08/May/2023:09:57:48 +0900] "GET /build/assets/touch-icon.2d6f6e49.png HTTP/1.1" 200 2248 "https://test...
2023-05-08T09:57:48.000+09:00	10.0.1.130 - campus [08/May/2023:09:57:48 +0900] "GET /build/assets/freedownload-icon.1a4192ba.png HTTP/1.1" 200 1292 "https...
2023-05-08T09:57:48.000+09:00	10.0.1.130 - campus [08/May/2023:09:57:48 +0900] "GET /build/assets/payment-icon.529c3a54.png HTTP/1.1" 200 1030 "https://te...

## 何とかしたい

- ・オペレーターさんに必要な情報を提供したい
- ・技術的な知識が少なくても解るような表示にしたい
- ・技術者にとっても内容を把握しやすいようにしたい
- ・費用は抑えたい。各案件への導入手間も抑えたい
- ・自社での運用に合わせて日々改良したい

※オペレーターさん = Web出願の電話サポートなどをする方たち。  
例えば「メールの送信状況を参照したい」ことが時々あります。



# ログ閲覧ツールを自作しました



## エンジニアの手間が減り、対応も早くなりました

例：

オペレーターさん「メールが届かないそうです」

エンジニア「メール送信ログを調べてみますね」

↓

オペレーターさん「メールが届かないそうなので、調べてみよう」

※もちろん内容によっては、改めてエンジニアが調査します。

## 仕組み

公式に用意された仕組みを利用して作成

1. Amazon CloudWatch Logs にログを転送
2. AWS SDKを使ってPHPプログラムから参照
3. 整形してブラウザの画面に表示

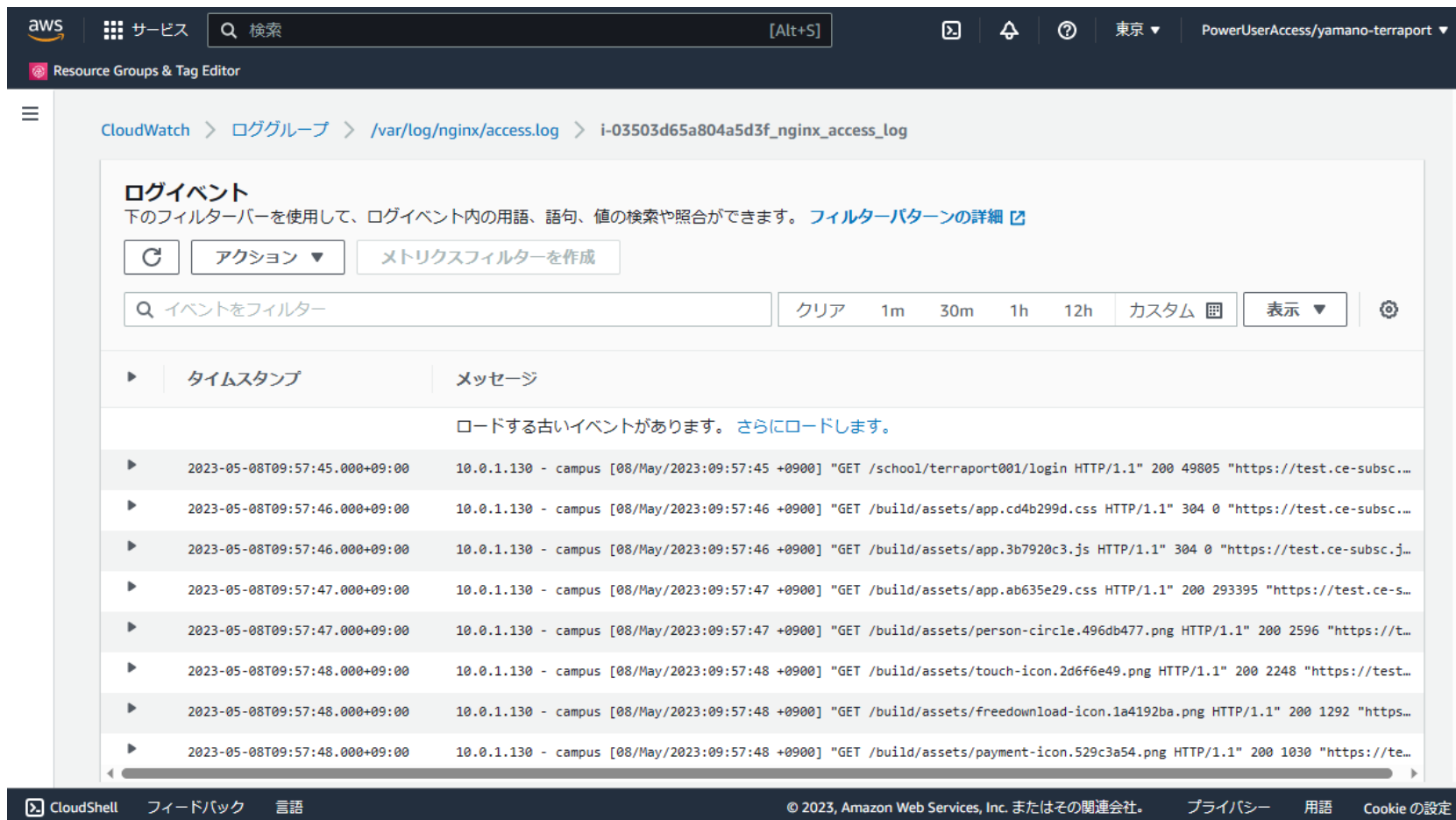
追加で工夫した点を紹介（以下3つの機能を後半で紹介）

1. 訪問者の移動経路と滞在時間を把握
2. リクエストやレスポンスタイムをグラフ化
3. メールの送信状況を把握

# 1. CloudWatch Logsにログを転送

※特別な手順は無いので、さらっと紹介します。

# 1のゴール = AWSコンソールにログを表示させる



The screenshot shows the AWS CloudWatch console interface. At the top, there's a navigation bar with the AWS logo, a search bar, and the user's profile. Below that, the breadcrumb trail indicates the current location: CloudWatch > ロググループ > /var/log/nginx/access.log > i-03503d65a804a5d3f\_nginx\_access\_log.

The main content area is titled "ログイベント" (Log Events). It includes a search bar with the text "イベントをフィルター" and a "クリア" button. There are also time range filters: "1m", "30m", "1h", "12h", and "カスタム". A "表示" (Display) button is also present.

Below the filters, there's a table with two columns: "タイムスタンプ" (Timestamp) and "メッセージ" (Message). The table shows a list of log events, each with a timestamp and a message. The messages are HTTP GET requests from 10.0.1.130 to various assets on a test server.

タイムスタンプ	メッセージ
2023-05-08T09:57:45.000+09:00	10.0.1.130 - campus [08/May/2023:09:57:45 +0900] "GET /school/terraport001/login HTTP/1.1" 200 49805 "https://test.ce-subsc...
2023-05-08T09:57:46.000+09:00	10.0.1.130 - campus [08/May/2023:09:57:46 +0900] "GET /build/assets/app.cd4b299d.css HTTP/1.1" 304 0 "https://test.ce-subsc...
2023-05-08T09:57:46.000+09:00	10.0.1.130 - campus [08/May/2023:09:57:46 +0900] "GET /build/assets/app.3b7920c3.js HTTP/1.1" 304 0 "https://test.ce-subsc.j..
2023-05-08T09:57:47.000+09:00	10.0.1.130 - campus [08/May/2023:09:57:47 +0900] "GET /build/assets/app.ab635e29.css HTTP/1.1" 200 293395 "https://test.ce-s...
2023-05-08T09:57:47.000+09:00	10.0.1.130 - campus [08/May/2023:09:57:47 +0900] "GET /build/assets/person-circle.496db477.png HTTP/1.1" 200 2596 "https://t...
2023-05-08T09:57:48.000+09:00	10.0.1.130 - campus [08/May/2023:09:57:48 +0900] "GET /build/assets/touch-icon.2d6f6e49.png HTTP/1.1" 200 2248 "https://test...
2023-05-08T09:57:48.000+09:00	10.0.1.130 - campus [08/May/2023:09:57:48 +0900] "GET /build/assets/freedownload-icon.1a4192ba.png HTTP/1.1" 200 1292 "https...
2023-05-08T09:57:48.000+09:00	10.0.1.130 - campus [08/May/2023:09:57:48 +0900] "GET /build/assets/payment-icon.529c3a54.png HTTP/1.1" 200 1030 "https://te...

At the bottom of the console, there's a footer with "CloudShell", "フィードバック", "言語", and "© 2023, Amazon Web Services, Inc. またはその関連会社。" along with links for "プライバシー", "用語", and "Cookie の設定".

## ロールを作成してAmazon EC2に紐づけ

セキュリティ認証情報 ➡ ロール ➡ ロールを作成

許可ポリシー	CloudWatchAgentServerPolicy
ロール名	CloudWatchAgentServerRole

対象のAmazon EC2に対して以下の操作を行う。

アクション ➡ セキュリティ ➡ IAMロールを変更

インスタンスID	対象のインスタンスIDが表示されていることを確認
IAMロール	CloudWatchAgentServerRole

## Amazon EC2に awslogs をインストール

```
$ sudo yum install -y awslogs
```

インストール後、設定ファイル `/etc/awslogs/awscli.conf` を編集。

```
[plugins]
cwlogs = cwlogs
[default]
#region = us-east-1
region = ap-northeast-1
```

東京リージョンを指定

## 設定ファイルで転送するログを指定

設定ファイル `/etc/awslogs/awslogs.conf` を編集。

```
[/var/log/messages]
datetime_format = %b %d %H:%M:%S
file = /var/log/messages
buffer_duration = 5000
log_stream_name = {instance_id}
initial_position = start_of_file
log_group_name = /var/log/messages
```

※デフォルトで `/var/log/messages` が転送されます。



## 設定ファイルで転送するログを指定

nginxのアクセスログとエラーログを追加で転送する例。

```
[/var/log/nginx/access.log]
datetime_format = %d/%b/%Y:%H:%M:%S
file = /var/log/nginx/access.log
buffer_duration = 5000
log_stream_name = {instance_id}
initial_position = start_of_file
log_group_name = /var/log/nginx/access.log
```

nginxのアクセスログを転送

参照するときの名前

```
[/var/log/nginx/error.log]
datetime_format = [%a %b %d %H:%M:%S %Y]
file = /var/log/nginx/error.log
buffer_duration = 5000
log_stream_name = {instance_id}
initial_position = start_of_file
log_group_name = /var/log/nginx/error.log
```

nginxのエラーログを転送

参照するときの名前

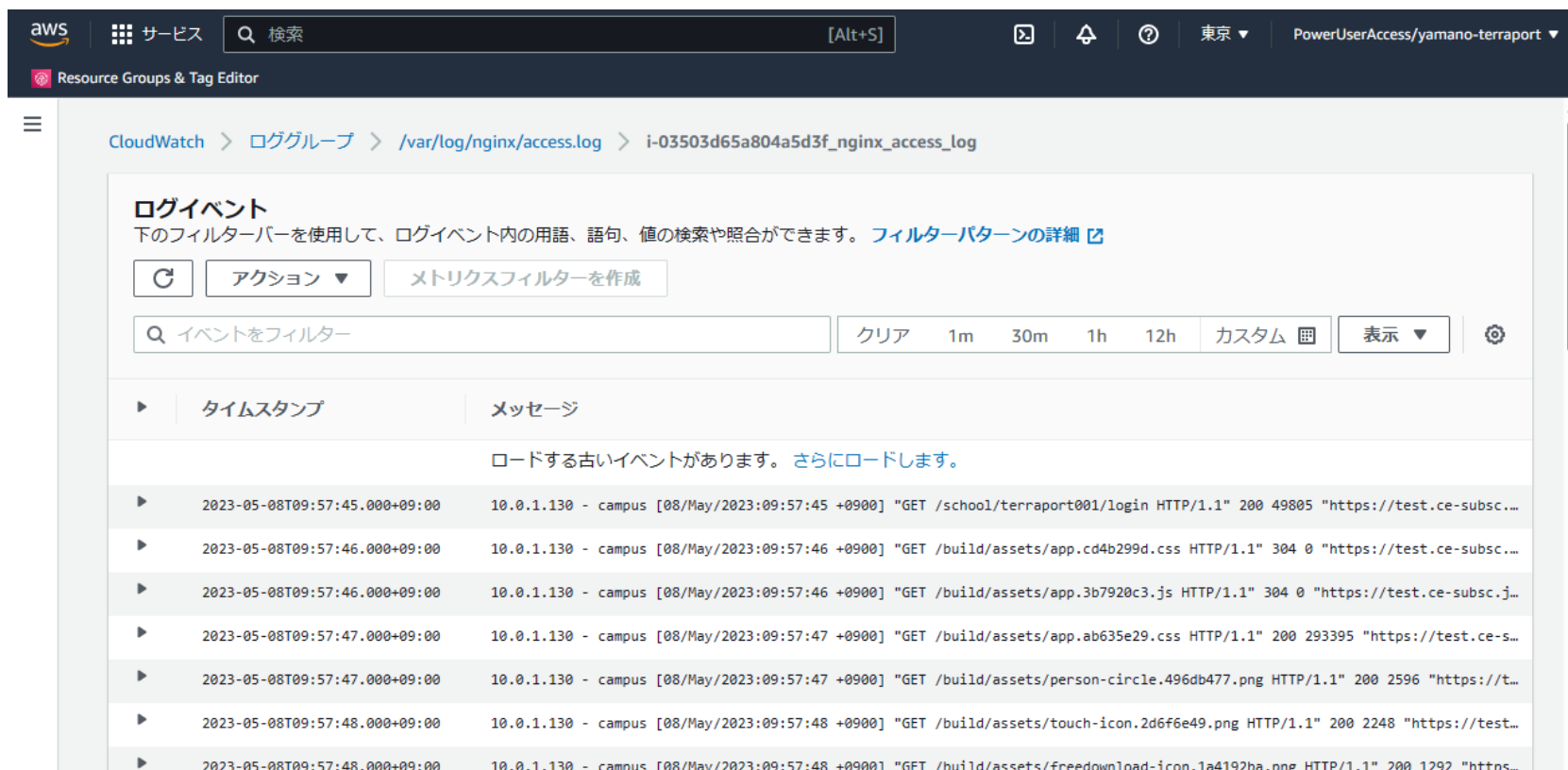
## CloudWatch Logsへの転送を開始

awslogsを起動 & 自動起動を設定。

```
$ sudo systemctl start awslogsd  
$ sudo systemctl enable awslogsd
```

# 設定完了

少し待つと、転送されたログをAWSコンソールで確認できる。



The screenshot shows the AWS CloudWatch console interface. The breadcrumb navigation indicates the path: CloudWatch > ロググループ > /var/log/nginx/access.log > i-03503d65a804a5d3f\_nginx\_access\_log. The main content area is titled "ログイベント" (Log Events) and includes a search bar with the text "イベントをフィルター" and a "表示" (Display) button. Below the search bar, there is a table with two columns: "タイムスタンプ" (Timestamp) and "メッセージ" (Message). The table contains several log entries, each with a timestamp and a message starting with "10.0.1.130 - campus [08/May/2023:09:57:45 +0900]".

タイムスタンプ	メッセージ
2023-05-08T09:57:45.000+09:00	10.0.1.130 - campus [08/May/2023:09:57:45 +0900] "GET /school/terraport001/login HTTP/1.1" 200 49805 "https://test.ce-subsc...
2023-05-08T09:57:46.000+09:00	10.0.1.130 - campus [08/May/2023:09:57:46 +0900] "GET /build/assets/app.cd4b299d.css HTTP/1.1" 304 0 "https://test.ce-subsc...
2023-05-08T09:57:46.000+09:00	10.0.1.130 - campus [08/May/2023:09:57:46 +0900] "GET /build/assets/app.3b7920c3.js HTTP/1.1" 304 0 "https://test.ce-subsc.j...
2023-05-08T09:57:47.000+09:00	10.0.1.130 - campus [08/May/2023:09:57:47 +0900] "GET /build/assets/app.ab635e29.css HTTP/1.1" 200 293395 "https://test.ce-s...
2023-05-08T09:57:47.000+09:00	10.0.1.130 - campus [08/May/2023:09:57:47 +0900] "GET /build/assets/person-circle.496db477.png HTTP/1.1" 200 2596 "https://t...
2023-05-08T09:57:48.000+09:00	10.0.1.130 - campus [08/May/2023:09:57:48 +0900] "GET /build/assets/touch-icon.2d6f6e49.png HTTP/1.1" 200 2248 "https://test...
2023-05-08T09:57:48.000+09:00	10.0.1.130 - campus [08/May/2023:09:57:48 +0900] "GET /build/assets/freedownload-icon.1a4192ba.png HTTP/1.1" 200 1292 "https...

## 2. AWS SDKを使ってPHPプログラムから参照

※このあたりからが本題です。

## 2のゴール = PHPでログを一覧表示させる

```
10.0.0.228 - example [29/May/2023:17:27:43 +0900] "GET /home HTTP/1.1" 200 3487 "-" "Mozilla/5.0 略 Safari/537.36" "203.0.113.1"  
10.0.0.228 - example [29/May/2023:17:27:43 +0900] "GET /css/style.css HTTP/1.1" 200 712 "https://example.com/home" "Mozilla/5.0 略 Safari/537.36" "203.0.113.1"  
10.0.0.228 - example [29/May/2023:17:27:43 +0900] "GET /js/script.css HTTP/1.1" 200 712 "https://example.com/home" "Mozilla/5.0 略 Safari/537.36" "203.0.113.1"
```

※ここでは「ログを一覧表示する」という例だけなので、上のログ内容は重要ではありません。

## アクセスキーを発行

セキュリティ認証情報 ➡ ユーザー ➡ ユーザーを追加

ユーザー名	monitor-user（任意の名前）
AWSマネジメントコンソールへのユーザーアクセス	提供しない
許可ポリシー	CloudWatchLogsReadOnlyAccess

さらに、上記ユーザーの詳細画面で

セキュリティ認証情報 ➡ アクセスキー

から **アクセスキーを作成** ボタンをクリックしてキーを発行。

## SDKの導入

AWSのデータを扱うための命令は、「AWS SDK」で提供されている。  
PHPの場合、AWS SDKはComposerからインストールできる。

```
$ composer require aws/aws-sdk-php
```

作成された `vendor/autoload.php` を読み込むことで、  
必要な命令を呼び出せるようになる。

```
<?php  
  
require_once '/path/to/vendor/autoload.php';
```

## プログラムを作成

以下のようなプログラムでログの内容を一覧表示できる。

```
use Aws\CloudWatchLogs\CloudWatchLogsClient;
```

```
$client = new CloudWatchLogsClient([  
    'credentials' => [  
        'key' => 'XXXXXXXXXX',  
        'secret' => 'XXXXXXXXXX',  
    ],  
    'region' => 'ap-northeast-1',  
    'version' => 'latest',  
]);
```

用意された命令を利用

発行したアクセスキー

(続く)



## プログラムを作成

以下のようなプログラムでログの内容を一覧表示できる。

(続き)

```
$result = $client->filterLogEvents([
    'logGroupName' => '/var/log/nginx/access.log',
    'startTime' => (time() - 60 * 60) * 1000,
    'endTime' => time() * 1000,
]);

foreach ($result->get('events') as $event) {
    echo $event['message'] . '<br>';
}
```

log\_group\_nameの値

開始時間と終了時間  
(UNIXタイムスタンプのミリ秒)

ログの内容を表示

※上記は /var/log/nginx/access.log から、直近60分の内容を一覧表示する指定です。

## ログの表示

以下のように表示できる。

```
10.0.0.228 - example [29/May/2023:17:27:43 +0900] "GET /home HTTP/1.1" 200 3487 "-" "Mozilla/5.0 略 Safari/537.36" "203.0.113.1"  
10.0.0.228 - example [29/May/2023:17:27:43 +0900] "GET /css/style.css HTTP/1.1" 200 712 "https://example.com/home" "Mozilla/5.0 略 Safari/537.36" "203.0.113.1"  
10.0.0.228 - example [29/May/2023:17:27:43 +0900] "GET /js/script.css HTTP/1.1" 200 712 "https://example.com/home" "Mozilla/5.0 略 Safari/537.36" "203.0.113.1"
```

## アクセスキーについて補足

認証情報のハードコーディングは推奨されないので、

- ・ 外部ファイルや環境変数から読み込む
- ・ EC2にIAMロールを割り当てる

などの対応を検討する。

参考: AWS SDK for PHPの認証情報管理

[https://d1.awsstatic.com/webinars/jp/pdf/services/2022\\_0831\\_AWS-Black-Belt\\_AWS\\_AWS\\_SDK\\_For\\_PHP\\_Auth\\_v1.pdf](https://d1.awsstatic.com/webinars/jp/pdf/services/2022_0831_AWS-Black-Belt_AWS_AWS_SDK_For_PHP_Auth_v1.pdf)

### 3. 整形してブラウザの画面に表示

## 画面に表示

ここまでの内容をベースに引き続きプログラムを書き、  
「絞り込み表示」「不要な情報を除外」「ログを見やすく装飾」  
などを行う。

## アクセスログ

Campusサブスク 検収環境 のログ /var/log/nginx/access.log は以下のとおりです。

日時

~

IPアドレス

ホスト / リクエスト

- 長いログを省略せずに表示  
 同一人物とみなして滞在時間を表示

検索する

日時	サーバ	IPアドレス	ホスト	リクエスト	結果	サイズ	リファラ
05/31 16:28:25	Campus-Staging-Web1	203.0.113.1	example.com	GET /entry/example	302	0KB	-
05/31 16:28:26	Campus-Staging-Web2	203.0.113.1	example.com	GET /entry/example/guidance	200	158KB	-
05/31 16:28:27	Campus-Staging-Web1	203.0.113.1	example.com	GET /_debugbar/assets/stylesheets?v=1676989262&the...	200	237KB	https://example.com/entry/terraportC
05/31 16:28:27	Campus-Staging-Web2	203.0.113.1	example.com	GET /_debugbar/assets/javascript?v=1676989262	200	363KB	https://example.com/entry/terraportC
05/31 16:28:32	Campus-Staging-Web1	203.0.113.1	example.com	GET /entry/example/guidance	200	158KB	-
05/31 16:28:32	Campus-Staging-Web1	203.0.113.1	example.com	GET /_debugbar/assets/javascript?v=1676989262	200	363KB	https://example.com/entry/terraportC
05/31 16:28:32	Campus-Staging-Web2	203.0.113.1	example.com	GET /_debugbar/assets/stylesheets?v=1676989262&the...	200	237KB	https://example.com/entry/terraportC

**完成！**

## 追加で工夫した点を紹介

引き続き、以下3つの機能を紹介します。

1. 訪問者の移動経路と滞在時間を把握
2. リクエストやレスポンスタイムをグラフ化
3. メールを送信状況を把握



# 1. 訪問者の移動経路と滞在時間を把握

## ログを解析して表示

表示対象でログを絞り込み、その内容を一覧にする



「訪問者の移動経路と、各ページの滞在時間」を知ることができる

	リクエスト	結果	リファラ	ユーザーエージェント	レスポンスタイム	滞在時間
sc.jp	GET /school/example/login	200	-	Windows 10 (or later) + Edge 114	0.0020ミリ秒	-
sc.jp	POST /school/example/login	302	https://test.ce-subsc.jp/school/example/login	Windows 10 (or later) + Edge 114	0.0010ミリ秒	10秒
sc.jp	GET /school/example/home	200	https://test.ce-subsc.jp/school/example/login	Windows 10 (or later) + Edge 114	0.0010ミリ秒	2秒
sc.jp	GET /school/example/exam/list	200	https://test.ce-subsc.jp/school/example/home	Windows 10 (or later) + Edge 114	0.0020ミリ秒	6秒

※滞在時間は単純に「遷移元ページのアクセス日時」と「遷移先ページのアクセス日時」の差分で求められます。

## 移動経路と滞在時間を把握

- ・ どのような経路で遷移したのか
- ・ どのページでどれくらい留まったのか

が判ると、

- ・ 意図されない手順でアクセスしていないか
- ・ 特定ページで滞在時間が長い = 何か操作に悩む部分があったのか

など、トラブル調査時の手助けになる。

## 2. リクエストやレスポンスタイムをグラフ化

## Webサーバの設定変更

Webサーバの設定を変更し、追加で必要なログを記録させる。

\$request_time	リクエスト処理時間 ※リクエストされてからログの書き込みが完了するまでの時間
\$http_x_forwarded_proto	訪問者が使用したプロトコル ※ロードバランサーを使用するため
\$host	ホスト名 ※サブドメインを割り当てた際に、どこに対するアクセスか把握するため

※nginxのlog\_formatの例です。

※上記の値は案件に関わらず常に追加で記録しています。

## Webサーバの設定変更

設定ファイル `/etc/nginx/nginx.conf` を編集。

```
'"$http_user_agent" "$http_x_forwarded_for";
```

↓

```
“"$http_user_agent” “$http_x_forwarded_for” $request_time (改行なし)  
$http_x_forwarded_proto $host';
```

編集したら設定を反映。

```
$ sudo service nginx reload
```

## Webサーバの設定変更

以下のようなログが追加で行末に記録されるようになる。

```
0.217 https abc.example.com  
0.002 https abc.example.com
```

リクエスト処理時間、訪問者が使用したプロトコル、ホスト名

## ログを解析して表示

リクエスト時間を記録したので、グラフ化すると以下を把握できる。

- ・ 「重い処理があったのか」 が一目で判る
- ・ 「リクエストの成功数」 「リクエストの失敗数」 の推移も一目で判る
- ・ 急に表示が重くなっていないか？ エラーが増えていないか？

※グラフは Chart.js ( <https://www.chartjs.org/> ) などを利用すれば、簡単に表示できます。  
※ 「リクエストの成功」 「リクエストの失敗」 は、アクセスログに元々記録されています。  
※ 「デプロイ直後に異常が発生していないか」 などの確認に使用できます。



Campusサブスク 検取環境 のログ 2023/06/08 14:52~2023/06/08 15:22 の解析結果は以下のとおりです。

## リクエスト

リクエスト数	226
秒間平均	0.1256

## エラー

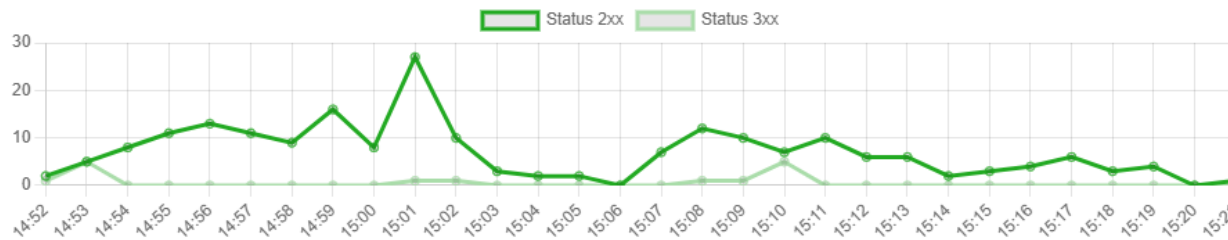
エラー数	3
発生率	1.33%

## レスポンス

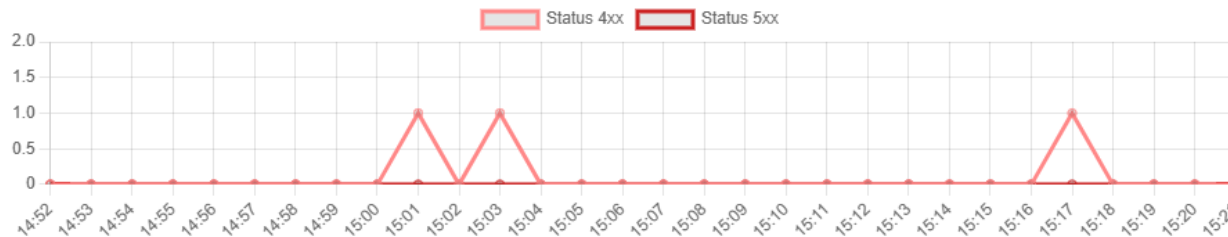
中央値	37.8075ミリ秒
平均値	57.1029ミリ秒
最大値	759.6540ミリ秒

## 推移

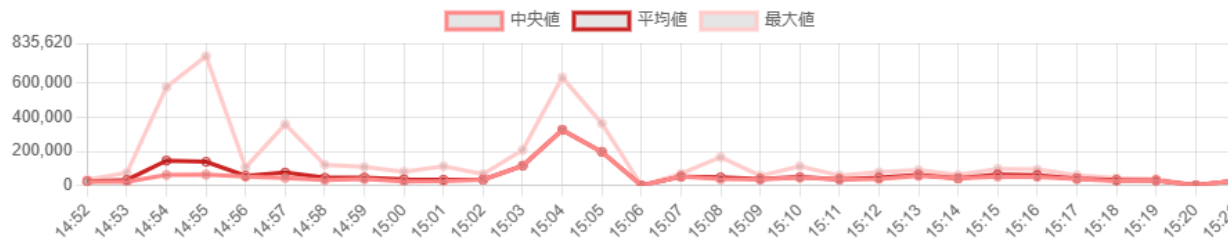
### リクエスト成功



### リクエスト失敗



### レスポンスタイム



### 3. メールの送信状況を把握

# Amazon SESのメール送信ログをS3に保存

## SESのメール送信ログを確認する方法

- A. Amazon Kinesis Data Firehose + Amazon OpenSearch Service
- B. Amazon Kinesis Data Firehose + Amazon S3

## 費用を抑えるためにBの方法を採用

メールが送信されると、送信ログがS3に保存される。

## Amazon SESのメール送信ログをS3に保存

メール送信ログの保存は、SESなどの設定で対応できる。

参考: SESの送信履歴を確認したい

<https://zenn.dev/isseeeeeey55/articles/61b350c27e1040>

参考: **【AWS】** AWS SESのログをKinesis Data Firehose経由でS3へ転送してみました - Qiita

<https://qiita.com/hyj624117615/items/6bf44f2f86c252e3e00b>




※「ログを表示する」という話から外れるので、ここでは参考ページを紹介するに留めます。

# Amazon SESのメール送信ログをS3に保存


以下のようにS3のバケットからメール送信ログを確認できる。

**オブジェクト (5)**

オブジェクトは、Amazon S3 に保存された基本的なエンティティです。Amazon S3 インベントリ [🔗](#)を使用して、バケット内のすべてのオブジェクトのリストを取得できます。他のユーザーが自分のオブジェクトにアクセスできるように、明示的にアクセス権限を付与する必要があります。詳細は[こちら](#) [🔗](#)

  S3 URI をコピー  URL をコピー  ダウンロード  開く  削除  アクション ▼  フォルダの作成  アップロード

🔍 プレフィックスでオブジェクトを検索 < 1 > ⚙️

<input type="checkbox"/>	名前 ▲	タイプ ▼	最終更新日時 ▼	サイズ ▼	ストレージクラス ▼
<input type="checkbox"/>	 campus-staging-ses-kinesis-1-2023-05-25-03-02-49-346d62f9-fb81-4ee8-b698-ae7c38ad63f7	-	2023/05/25 12:03:52 PM JST	11.3 KB	スタンダード
<input type="checkbox"/>	 campus-staging-ses-kinesis-1-2023-05-25-03-03-52-116647ce-b009-4e73-9a1a-7179e0c00ca3	-	2023/05/25 12:04:55 PM JST	3.6 KB	スタンダード
<input type="checkbox"/>	 campus-staging-ses-kinesis-1-2023-05-25-03-05-21-19d07c90-0656-496e-aaf7-61a752ad6b27	-	2023/05/25 12:06:23 PM JST	3.7 KB	スタンダード
<input type="checkbox"/>	 campus-staging-ses-kinesis-1-2023-05-25-03-06-32-a42d471b-edc0-4c72-9683-f7f7b2091353	-	2023/05/25 12:07:35 PM JST	7.6 KB	スタンダード
<input type="checkbox"/>	 campus-staging-ses-kinesis-1-2023-05-25-03-15-32-09ee347a-32a5-4b69-9fab-7f1d6533552f	-	2023/05/25 12:16:34 PM JST	3.7 KB	スタンダード

## Amazon SESのメール送信ログをプログラムで取得

- ・ ログの内容を1つずつ確認するのは大変
- ・ 読み解くには知識が必要



メール送信ログについても、判りやすい画面を提供できるように。

## 許可ポリシーを追加

先ほど作成した「CloudWatch Logsのデータを参照する」ユーザーに、さらに許可ポリシーを追加。

先ほど追加した許可ポリシー	CloudWatchLogsReadOnlyAccess
新たに追加する許可ポリシー	AmazonS3ReadOnlyAccess

## Amazon SESのメール送信ログをプログラムで取得

以下のようなプログラムでファイル内容を表示できる。

```
use Aws::S3::S3Client;

$client = new S3Client([
  'credentials' => [
    'key' => 'XXXXXXXXXX',
    'secret' => 'XXXXXXXXXX',
  ],
  'region' => 'ap-northeast-1',
  'version' => 'latest',
]);
```

用意された命令を利用

発行したアクセスキー

(続く)



## Amazon SESのメール送信ログをプログラムで取得

以下のようなプログラムでファイル内容を表示できる。

(続き)

```
$result = $client->getObject([  
    'Bucket' => 'example',  
    'Key' => 'log/test.txt',  
]);
```

表示するバケット

表示するファイル

```
header('Content-Type: text/plain');  
echo $result['Body'];
```

ファイルの内容を表示

## Amazon SESのメール送信ログをプログラムで取得

以下のように、メールの送信ログを一覧表示できる。

送信日時	送信元	送信先	件名	追加情報	結果
05/24 17:40:25	no-reply@test.ce-subsc.jp (Campusサブスク(検収))	admin@example.com	【テラポート学園】出願・エントリー申請を受け付けました (管理者向け)	-	成功
05/24 17:40:24	no-reply@test.ce-subsc.jp (テラポート学園)	taro.yamada@example.com	【テラポート学園】出願・エントリー申請を受け付けました	-	成功
05/24 17:45:01	no-reply@test.campus-s.jp (テラポート学園)	taro.yamada@example.com	【テラポート学園】検定料 入金を確認しました	-	成功
05/24 18:02:19	no-reply@test.ce-subsc.jp (テラポート学園)	taro.yamada@example.com	【テラポート学園】受験票が発行されました	-	成功

## メール送信ログを装飾して表示

さらに以下の対応を行う。

- ・ エラーメールは強調して表示
- ・ エラーになった理由もすぐに参照できるように  
(エラーコードをもとに、エラーの理由を日本語で表示)

## メール送信ログを装飾して表示

エラーメールの表示例。

	件名	追加情報	結果
/amada@example.com	【テラポート学園】出願・エントリー申請を受け付けました	-	成功
example.com	【テラポート学園】出願・エントリー申請を受け付けました（管理者向け）	-	成功
jki@example.com	【テラポート学園】仮登録が完了しました	<a href="#">バウンスメール検索</a>	エラー: 受信ボックスのストレージ不足。

※CloudWatch Logsの話からは外れますが、  
別途バウンスメールも解析して本システムから参照できるようにしています。

## ツールの歴史

「nginxのアクセスログとエラーログを一覧して表示する」  
だけの、1~2時間程度で作った簡易ツール



使いながら徐々に改良



案件の内容や規模に関わらず、同じ操作性で検索&閲覧などできる



運用負荷を減らすことができた

# まとめ

## まとめ

- ・ CloudWatch Logs で、容易にログの集中管理ができます
- ・ AWS SDK が充実しており、各種プログラミング言語から扱えます
- ・ 社内の運用に合わせて、ちょっとしたツールを容易に作成できます

中小規模な案件やスモールスタートな案件がいくつもある場合、ご参考になれば幸いです。

ありがとうございました