

AWS

S U M M I T

# AWS の運用監視入門

Amazon CloudWatch

技術統括本部 ソリューションアーキテクト  
藤倉 和明

2017年6月2日



# 本セッションのFeedbackをお願いします

受付でお配りしたアンケートに本セッションの満足度やご感想などをご記入ください  
アンケートをご提出いただきました方には、もれなく**素敵なAWSオリジナルグッズ**を  
プレゼントさせていただきます



アンケートは受付、パミール3FのEXPO展示会場内にて回収させていただきます

# 自己紹介

藤倉 和明 (ふじくら かずあき)

アマゾン ウェブ サービス ジャパン株式会社  
エンタープライズソリューション部  
ソリューションアーキテクト

好きなAWSサービス

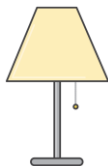
- Amazon Virtual Private Cloud (VPC)
- Amazon CloudWatch



# 本日皆様にお持ち帰りして頂く内容



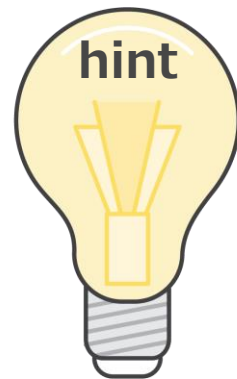
運用をもっと簡単に



夜ぐっすり眠れる運用体制へ



運用もスケーラブルに



# 本セッションについて



- 想定するオーディエンス
  - オンプレミス、クラウド問わず**システム運用**をしている人
  - システム運用、監視に**課題を感じている人**
- セッションの概要
  - **フルマネージド**運用監視サービス **Amazon CloudWatch**を中心とした、**クラウドの運用**についてご紹介

# クラウドの運用

# AWSのクラウド運用



- ・ コスト最適化
- ・ Well Architected



- ・ 監視
- ・ ログ管理
- ・ 自動化



- ・ 標準化
- ・ テンプレート



クラウド運用



- ・ API管理
- ・ 監査



- ・ 構成管理、変更管理
- ・ コンプライアンス強化



# AWSのクラウド運用

今日はここを中心に



- ・ コスト最適化
- ・ Well Architected



- ・ 監視
- ・ ログ管理
- ・ 自動化



- ・ 標準化
- ・ テンプレート



クラウド運用



- ・ API管理
- ・ 監査



- ・ 構成管理、変更管理
- ・ コンプライアンス強化





# システム監視とは



# システム監視とは

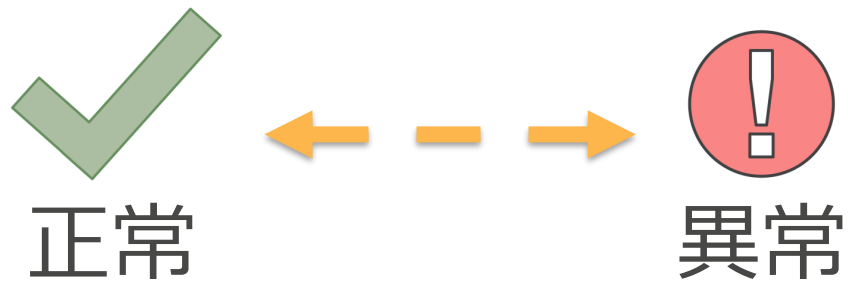


正常



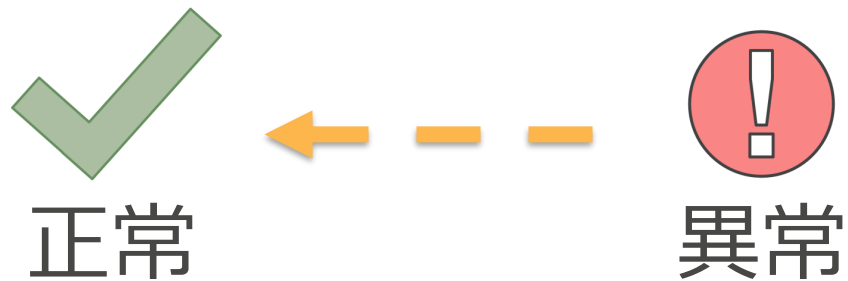
異常

# システム監視とは



- 「正常な状態」を定義
- 「異常な状態」を定義
- その状態の変化

# 障害対応：「異常な状態」を「正常な状態」へ



障害対応は重要な「運用」

- 予め決まっている対応
- 不測の事態
  - まずは落ち着いて
  - 一時復旧、根本対応
  - 再発防止

# クラウドにおける「正常」「異常」とは？

**Serverless ?**

**スケールア  
ウト？**

**データレイク？**

**ビッグデータ？**

**マネージドサー  
ビス？**



**AWSクラウドにおけるシステム運用監視は  
CloudWatchが最適！**



**CloudWatch**

# Amazon CloudWatchとは

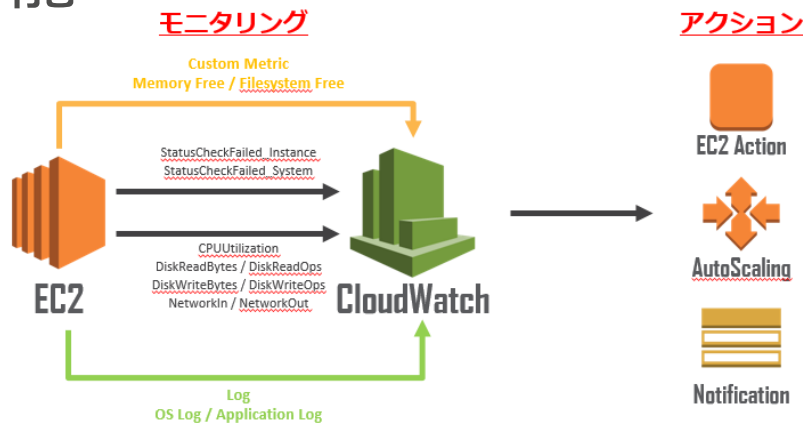


## AWSの各種リソースを監視する マネージドサービス

- **セットアップ不要**
- 正常な状態を継続的に監視
- 異常な状態の検知と、自動的な復旧をサポート
- 各メトリックスをベースとしたアラーム(通知)、アクションの設定が可能

## 多くのAWSサービスの監視が可能

- Amazon EC2
- Amazon EBS
- Amazon RDS
- Elastic Load Balancing など



# Amazon CloudWatchのできる事

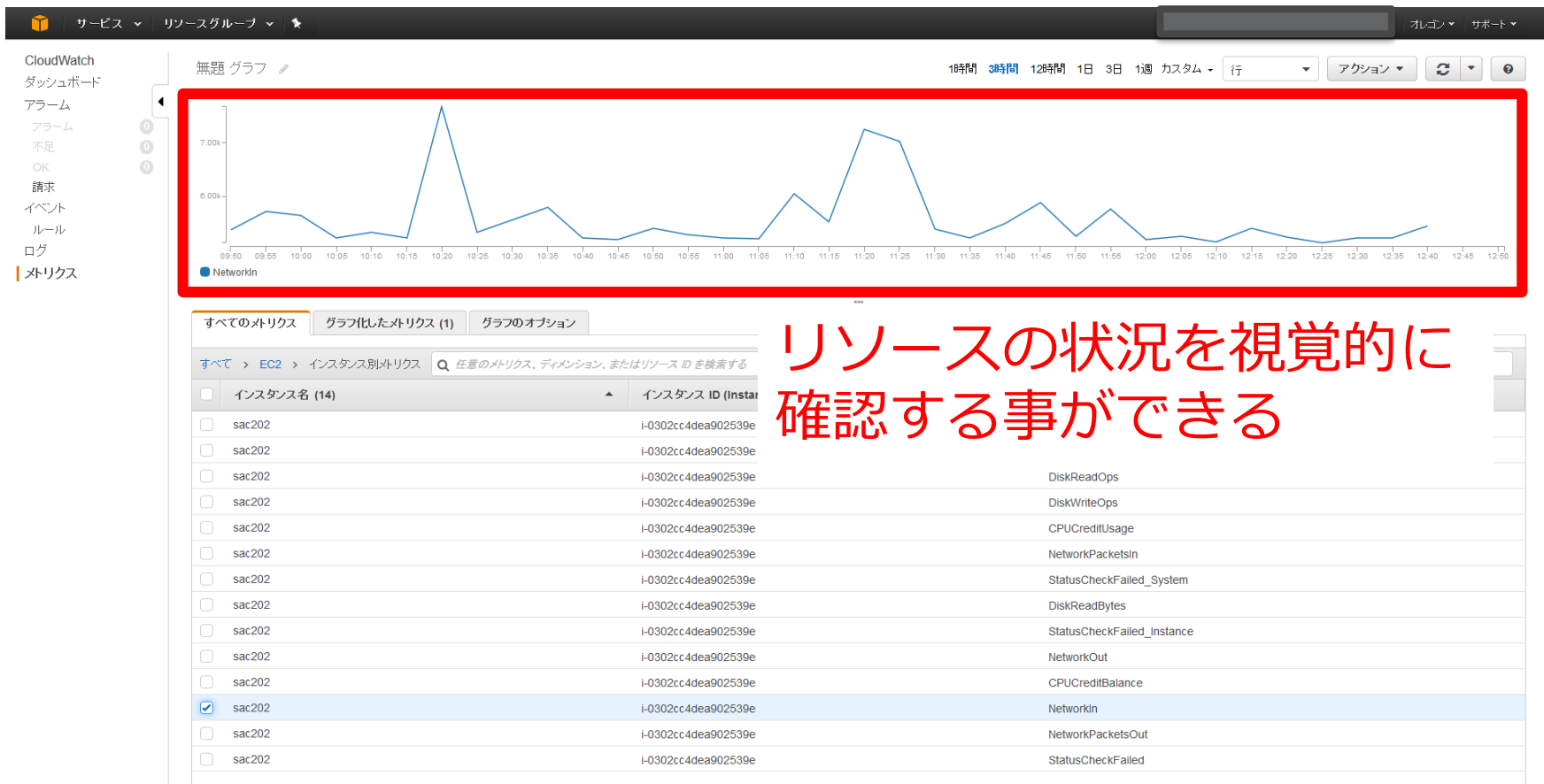


CloudWatch

- CloudWatch
  - システム監視サービス
    - ✓ 死活監視 / 性能監視 / キャパシティ監視
- CloudWatch Logs
  - ログ管理プラットフォームサービス
    - ✓ EC2上のOS, APPのログ
    - ✓ AWSマネージド サービスのログ
- CloudWatch Events
  - AWS上リソースの状態監視サービス
  - AWSリソースに対するイベントをトリガーにアクションを実行する機能



# CloudWatch利用イメージ



リソースの状況を視覚的に  
確認することができる

# CloudWatchのメトリックス

## 標準メトリックス (EC2)

CPUUtilization  
CPUCreditBalance  
CPUCreditUsage  
DiskReadBytes  
DiskWriteBytes  
DiskReadOps  
DiskWriteOps  
NetworkOut  
NetworkIn  
NetworkPacketsIn  
NetworkPacketsOut  
StatusCheckFailed Instance  
StatusCheckFailed  
StatusCheckFailed System  
BurstBalance

## カスタムメトリックス

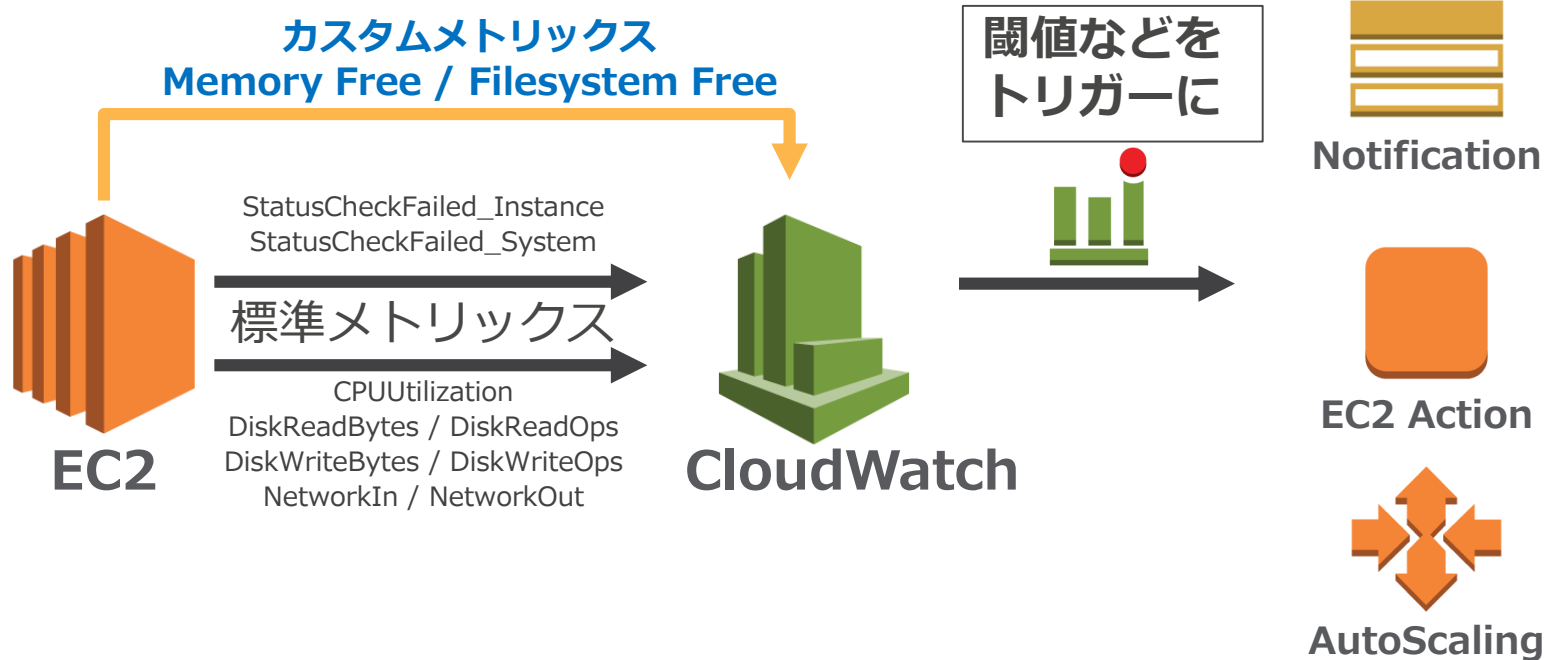
標準メトリックスでは  
収集できないメトリックス



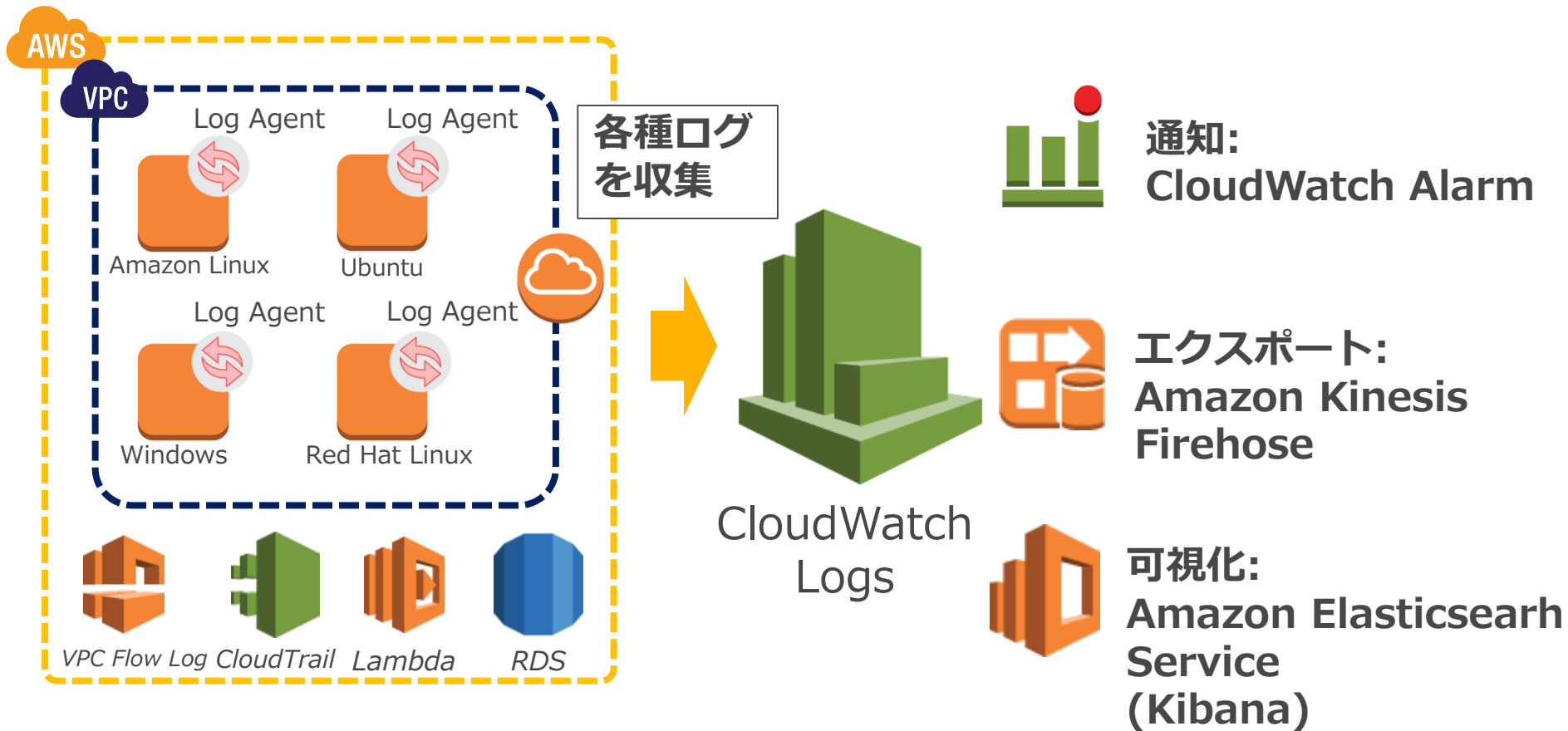
# Amazon CloudWatch のアクション機能

## モニタリング

## アクション



# CloudWatch Logs利用イメージ



# ログモニタリングイメージ

ログ内容はタイムスタンプとログメッセージ（UTF-8）で構成

The screenshot shows the AWS CloudWatch console interface. The breadcrumb navigation indicates the path: CloudWatch > ロググループ > Default-Log-Group > i-0299505423feee84b. The left sidebar shows the 'CloudWatch' section with 'ダッシュボード' selected. The main area displays a log stream with a filter bar at the top. A red box highlights the filter bar, with the text 'フィルター（検索）' (Filter (Search)) written in red. Below the filter bar, the log stream is displayed with columns for 'タイムスタンプ' (Timestamp) and 'ログメッセージ' (Log Message). A red box highlights the first five log entries, with the text 'タイムスタンプ' (Timestamp) and 'ログメッセージ' (Log Message) written in red. The log entries include system messages such as 'The Software Protection service entered the running state.', 'The Device Setup Manager service entered the running state.', and 'The WinHTTP Web Proxy Auto-Discovery Service service entered the running state.'

CloudWatch

サービス ▾ リソースグループ ▾ ☆

CloudWatch > ロググループ > Default-Log-Group > i-0299505423feee84b

フィルター（検索）

イベントのフィルター

すべて展開 行 テキスト

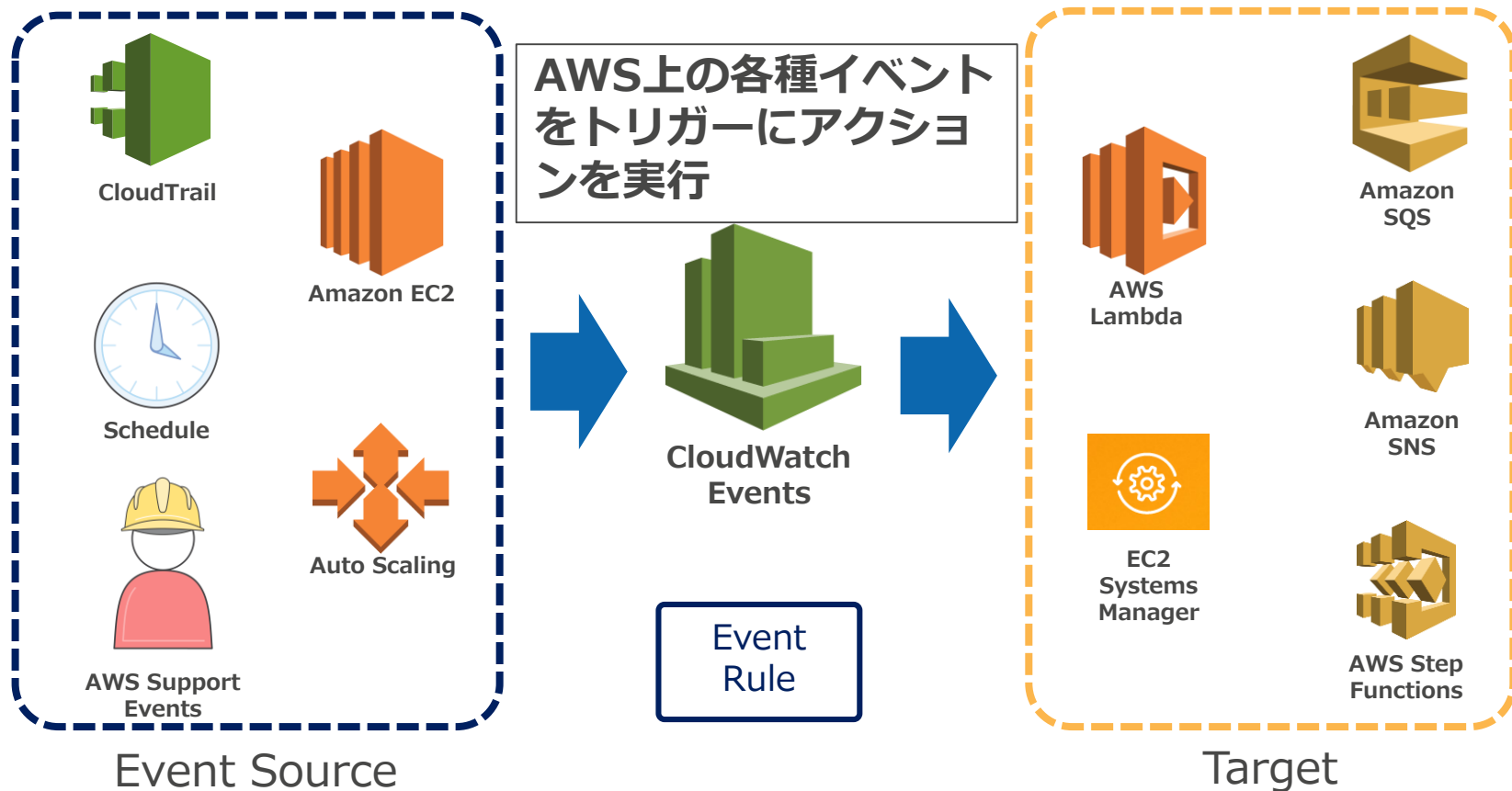
すべて 30秒 5分 1時間 6時間 1日 1週 カスタム

タイムスタンプ

ログメッセージ

タイムスタンプ	ログメッセージ
14:57:16	[System] [Information] [7036] [Service Control manager] [WIN-3AM8GUMMNLG.defaultvc-adtest2.local] [The Software Protection service entered the stopped state.]
15:17:32	[System] [Error] [1111] [Microsoft-Windows-TerminalServices-Printers] [WIN-3AM8GUMMNLG.defaultvc-adtest2.local] [Driver Microsoft Print To PDF required for printer Microsoft Print To PDF]
15:17:33	[System] [Information] [7036] [Service Control Manager] [WIN-3AM8GUMMNLG.defaultvc-adtest2.local] [The Device Setup Manager service entered the running state.]
15:17:35	[System] [Error] [1111] [Microsoft-Windows-TerminalServices-Printers] [WIN-3AM8GUMMNLG.defaultvc-adtest2.local] [Driver Send to Microsoft OneNote 16 Driver required for printer Send to Microsoft OneNote 16]
15:19:35	[System] [Information] [7036] [Service Control Manager] [WIN-3AM8GUMMNLG.defaultvc-adtest2.local] [The Device Setup Manager service entered the stopped state.]
15:21:09	[System] [Information] [7036] [Service Control Manager] [WIN-3AM8GUMMNLG.defaultvc-adtest2.local] [The WinHTTP Web Proxy Auto-Discovery Service service entered the stopped state.]
15:34:57	[System] [Error] [36888] [Schannel] [WIN-3AM8GUMMNLG.defaultvc-adtest2.local] [A fatal alert was generated and sent to the remote endpoint. This may result in termination of the connection.]
15:34:57	[System] [Error] [36888] [Schannel] [WIN-3AM8GUMMNLG.defaultvc-adtest2.local] [A fatal alert was generated and sent to the remote endpoint. This may result in termination of the connection.]
15:40:04	[System] [Error] [36888] [Schannel] [WIN-3AM8GUMMNLG.defaultvc-adtest2.local] [A fatal alert was generated and sent to the remote endpoint. This may result in termination of the connection.]
15:40:04	[System] [Error] [36888] [Schannel] [WIN-3AM8GUMMNLG.defaultvc-adtest2.local] [A fatal alert was generated and sent to the remote endpoint. This may result in termination of the connection.]
15:54:42	[System] [Information] [7036] [Service Control Manager] [WIN-3AM8GUMMNLG.defaultvc-adtest2.local] [The WinHTTP Web Proxy Auto-Discovery Service service entered the running state.]

# CloudWatch Events利用イメージ



# 例：コンソールサインイン時にアラートを飛ばす



イベントパターン ⓘ

サービス別のイベントに

サインインイベント

サービス名 AWS コンソールのサインイン

イベントタイプ サインインイベント

☐ 任意のユーザー ☒ ARN 別の特定のユーザー

arn:aws:iam::029845397858:user/kazuakf



イベントがイベントパターンに一致したときに呼び出すターゲットを選択します。

SNSトピック

SNSトピック

トピック\* alert-iam

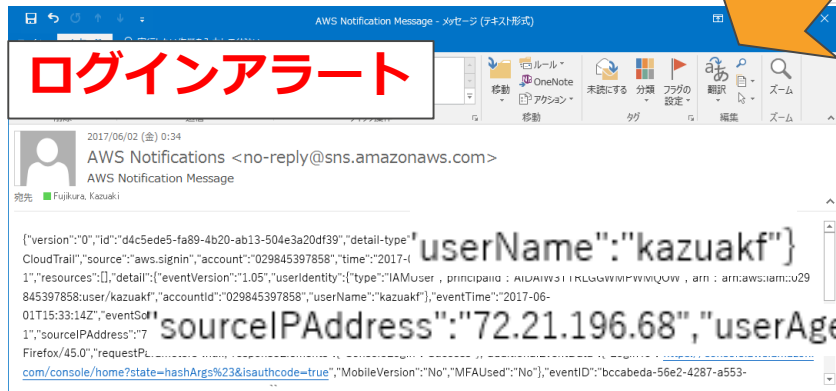
▼ 入力 の 設定

☒ 一致したイベント ⓘ

☐ 一致したイベントの一部 ⓘ

☐ 定数 (JSON テキスト) ⓘ

☐ インプットトランスフォーマー ⓘ



# Amazon CloudWatchのできる事



CloudWatch

- CloudWatch
  - システム監視サービス
    - ✓ 死活監視 / 性能監視 / キャパシティ監視
- CloudWatch Logs
  - ログ管理プラットフォームサービス
    - ✓ EC2上のOS, APPのログ
    - ✓ AWSマネージド サービスのログ
- CloudWatch Events
  - AWS上リソースの状態監視サービス
  - AWSリソースに対するイベントをトリガーにアクションを実行する機能



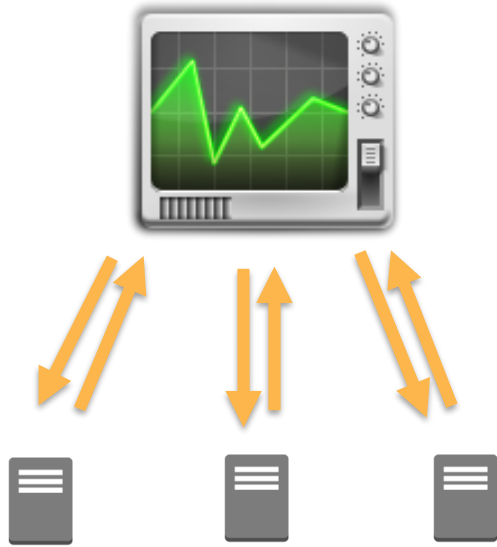
# Why CloudWatch

1. Pollingモデルから、**Pushモデル**へ
2. サーバ中心の監視から、**サービスの監視**へ
3. AWSサービスとの**連携**

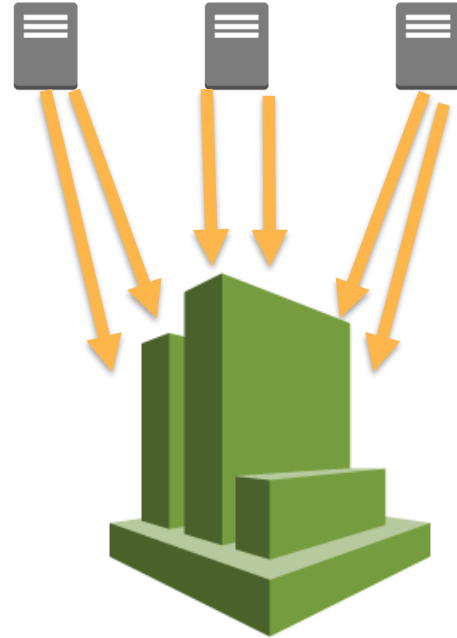
# Why CloudWatch

1. Pollingモデルから、**Pushモデル**へ
2. サーバ中心の監視から、**サービスの監視**へ
3. AWSサービスとの**連携**

# Polling / Push

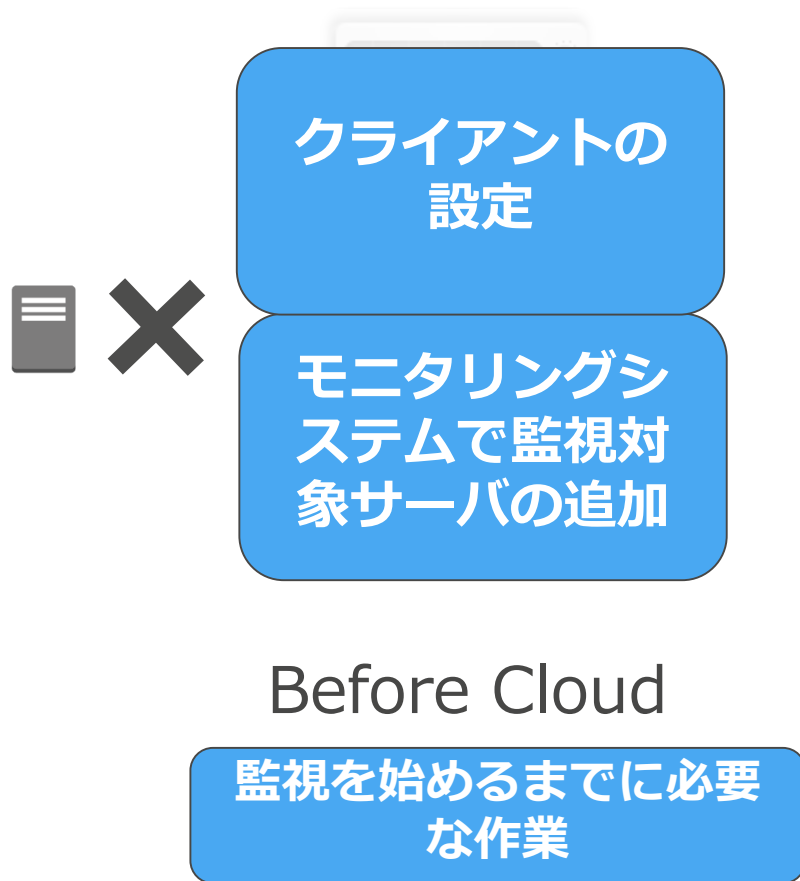


Before Cloud

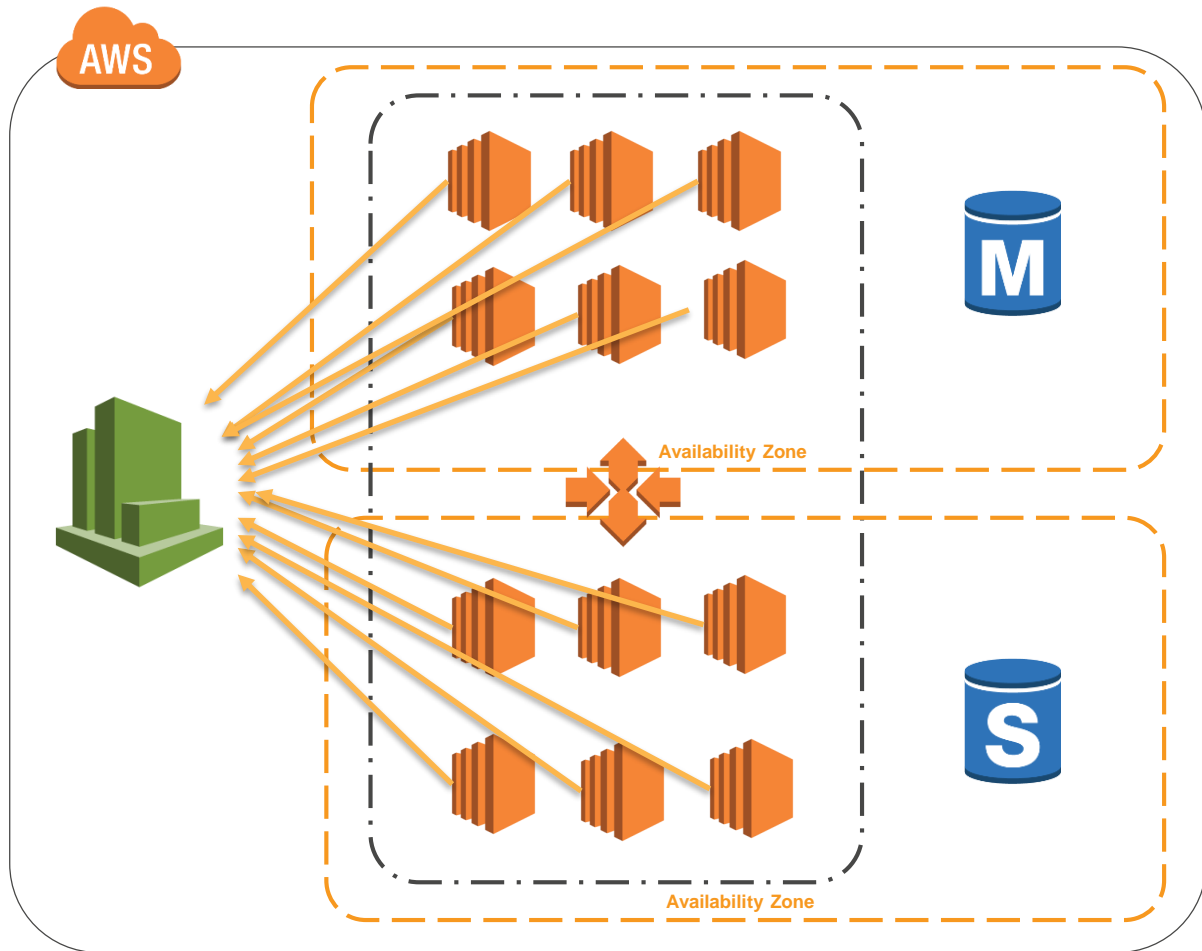


Amazon CloudWatch

# Polling / Push



# その監視、オートスケーलに対応していますか？

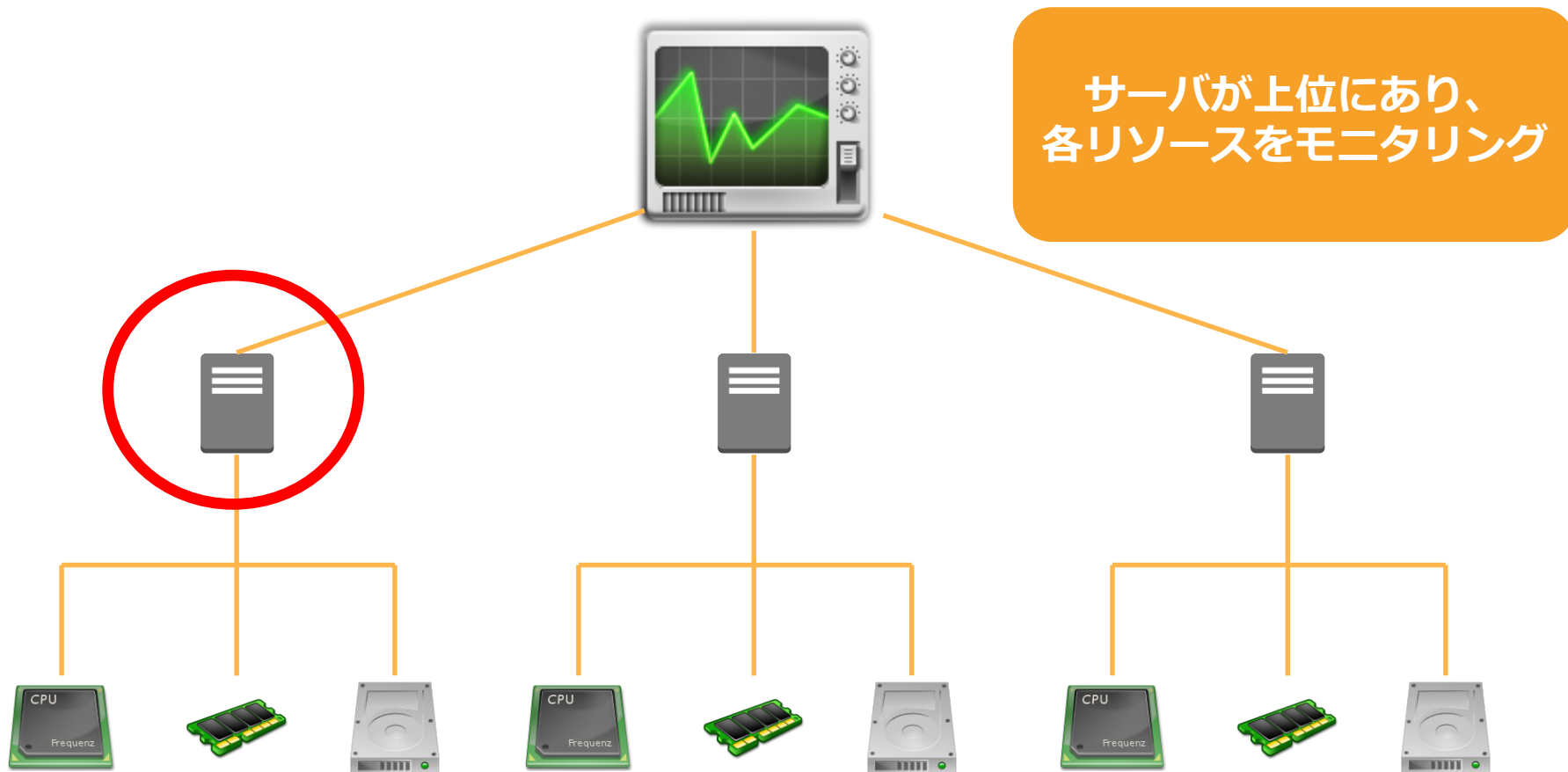


- ダイナミックに変化するサーバの監視は、従来のPollingモデルは難しかった
- Pushモデルの監視ソリューションでは、オートスケーल等のダイナミックに増減するサーバにも対応可能

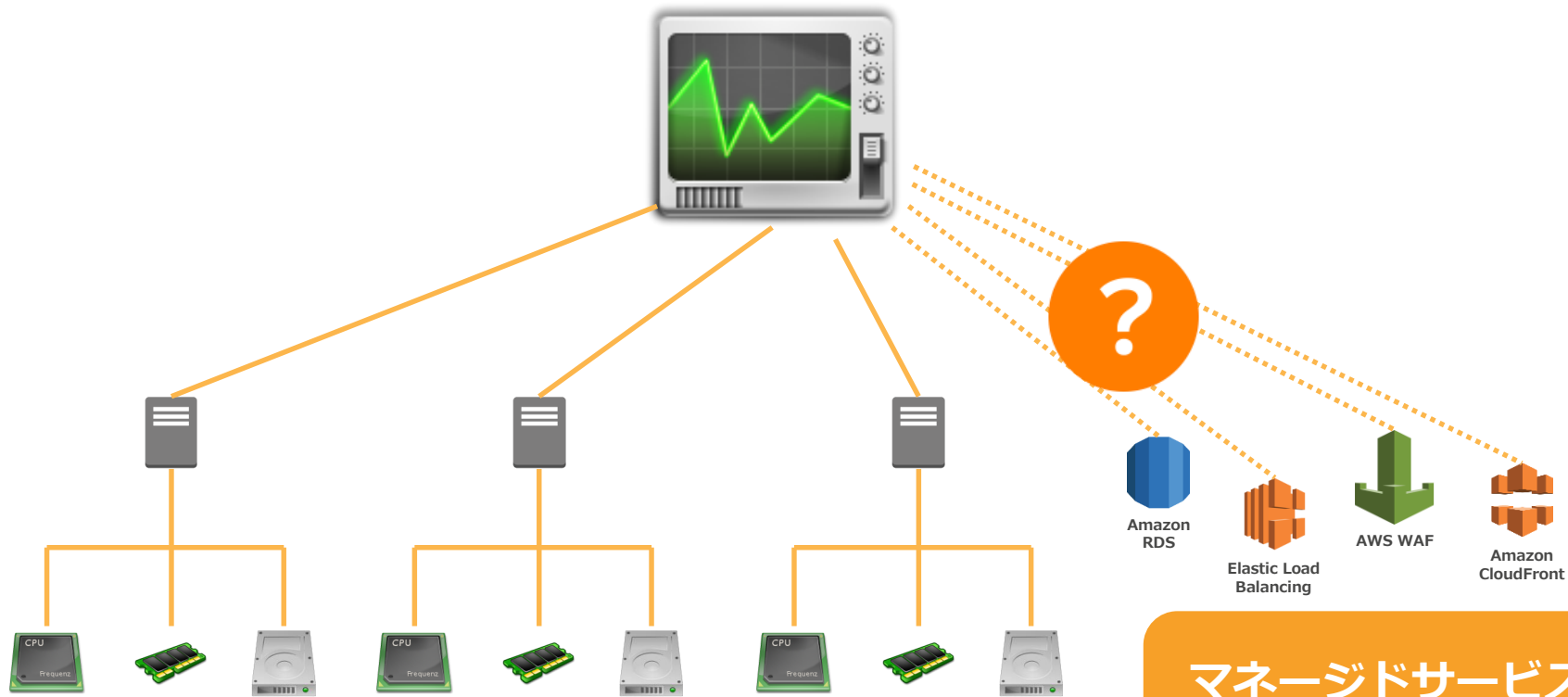
# Why CloudWatch

1. Pollingモデルから、Pushモデルへ
2. サーバ中心の監視から、**サービスの監視**へ
3. AWSサービスとの連携

# 従来のモニタリングシステム



# 従来のモニタリングシステム



マネージドサービスの  
モニタリングは？



# マネージドサービスの場合

アプリケーション最適化	アプリケーション最適化	アプリケーション最適化
スケール	スケール	スケール
高可用性	高可用性	高可用性
バックアップ	バックアップ	バックアップ
MW パッチ適用	MW パッチ適用	MW パッチ適用
MW インストール	MW インストール	MW インストール
OS パッチ適用	OS パッチ適用	OS パッチ適用
OS インストール	OS インストール	OS インストール
サーバー管理	サーバー管理	サーバー管理
ラッキング	ラッキング	ラッキング
電源, 空調, ネットワーク	電源, 空調, ネットワーク	電源, 空調, ネットワーク
オンプレミス	On EC2	マネージドサービス



# マネージドサービスの場合

アプリケーション最適化	アプリケーション最適化	アプリケーション最適化
スケール	スケール	スケール
高可用性	高可用性	高可用性

パッチ適用 / バックアップ / スケールまでお任せの  
マネージドサービスを活用する事で  
**より付加価値の高い仕事に集中**できる  
クラウドの運用にマネージドサービスの利用は不可欠

サーバー管理	サーバー管理	サーバー管理
ラッキング	ラッキング	ラッキング
電源, 空調, ネットワーク	電源, 空調, ネットワーク	電源, 空調, ネットワーク

オンプレミス

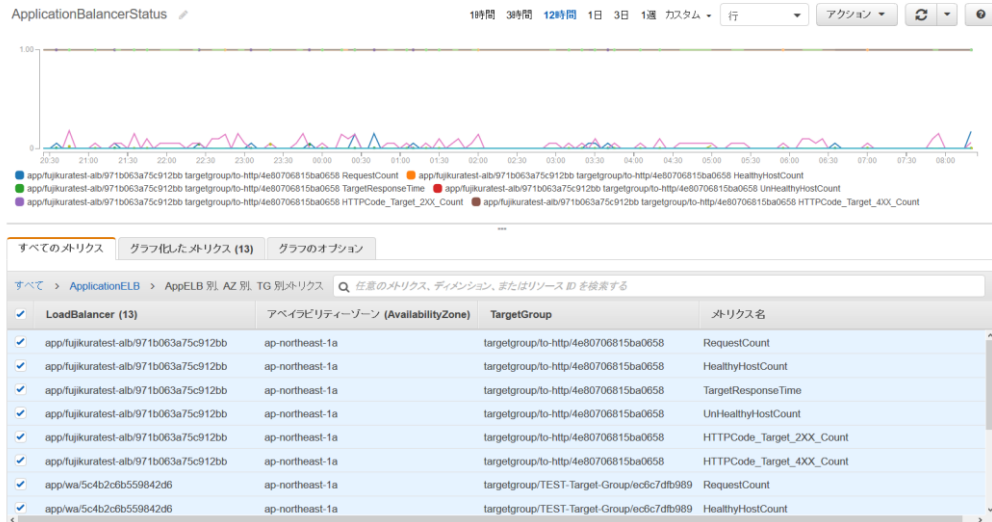
On EC2

マネージドサービス

# マネージドサービスの活用にはCloudWatch



40を超えるマネージドサービスが追加の設定  
無くメトリクスの収集が可能  
サーバ中心の監視から**サービス中心の監視**へ  
シフトする事で運用もスケラブルに



# CloudWatchに対応するAWSサービス

AWS サービス	名前空間	AWS サービス	名前空間
Amazon API Gateway	AWS/ApiGateway	Amazon EMR	AWS/ElasticMapReduce
Auto Scaling	AWS/AutoScaling	AWS IoT	AWS/IoT
AWS Billing	AWS/Billing	AWS Key Management Service	AWS/KMS
Amazon CloudFront	AWS/CloudFront	AWS Kinesis Data Analytics	AWS/KinesisAnalytics
Amazon CloudSearch	AWS/CloudSearch	AWS Kinesis Data Firehose	AWS/Firehose
Amazon CloudWatch Events	AWS/Events	AWS Kinesis Data Streams	AWS/Kinesis
Amazon CloudWatch Logs	AWS/Logs	AWS Lambda	AWS/Lambda
AWS Database Migration Service		AWS Machine Learning	AWS/ML
Amazon DynamoDB		AWS OpsWorks	AWS/OpsWorks
Amazon EC2		AWS Polly	AWS/Polly
Amazon EC2 Container Instance		AWS Redshift	AWS/Redshift
AWS Elastic Beanstalk			AWS/RDS
Amazon Elastic Block Store			AWS/Route53
Amazon Elastic File System		AWS Shield Advanced	AWS/DDoSProtection
Elastic Load Balancing		Amazon Simple Email Service	AWS/SES
Elastic Load Balancing (Application Load Balancers)		Amazon Simple Notification Service	AWS/SNS
Amazon Elastic Transcoder	AWS/ElasticTranscoder	Amazon Simple Queue Service	AWS/SQS
Amazon ElastiCache	AWS/ElastiCache	Amazon Simple Storage Service	AWS/S3
Amazon Elasticsearch Service	AWS/ES	Amazon Simple Workflow Service	AWS/SWF
		AWS Storage Gateway	AWS/StorageGateway
		AWS WAF	AWS/WAF
		Amazon WorkSpaces	AWS/WorkSpaces

40を超える  
AWSサービスに対応

# Why CloudWatch

1. Pollingモデルから、**Pushモデル**へ
2. サーバ中心の監視から、**サービスの監視**へ
3. AWSサービスとの**連携**

# AWSサービスとの連携



Any Services



予め定めてある障害対応手順は、サービス連携により  
**運用の自動化**が可能



Notification



EC2 Action



AutoScaling



Amazon  
S3 Export



Kinesis



Amazon  
Elasticsearch  
Service



Amazon  
SQS



Amazon  
SNS



Lambda



EC2 Systems  
Manager



AWS Step  
Functions

CloudWatch

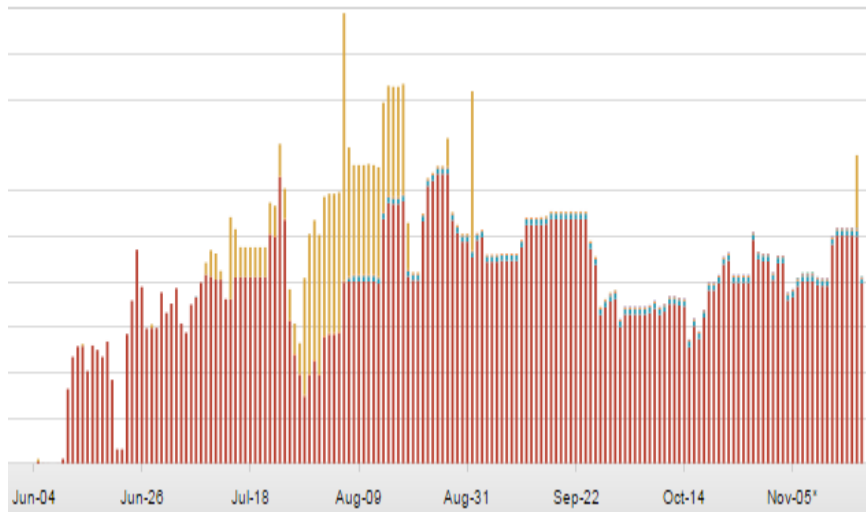
CloudWatch Logs

CloudWatch Events

# クラウドならではの監視も

## Billingアラーム設定

- 課金状況をCloudWatch監視
- 一定金額を超えるとアラームメール通知が可能
- アラームの設定はVirginiaリージョンから設定



### Alarm Threshold

Provide the details and threshold for your alarm. Use the graph on the right to help set the appropriate threshold.

Name: Billing Alarm

Description: AWS Billing Alarm

Whenever charges for: EstimatedCharges

is:  $\geq$  USD \$ 100

### Actions

Define what actions are taken when your alarm changes state.

Notification

Delete

Whenever this alarm: State is ALARM

Send notification to: Select a notification list

New list Enter list

+ Notification

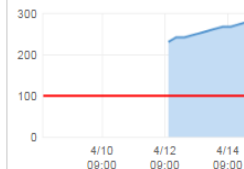
+ AutoScaling Action

+ EC2 Action

### Alarm Preview

This alarm will trigger when the blue line goes up to or above the red line

EstimatedCharges  $\geq$  100



Namespace: AWS/Billing

Currency: USD

Metric Name: EstimatedCharges

# (再掲) Why CloudWatch

1. Pollingモデルから、**Pushモデル**へ
2. サーバ中心の監視から、**サービスの監視**へ
3. AWSサービスとの**連携**



# Dive Deep CloudWatch

# (再掲) Amazon CloudWatchのできる事



CloudWatch

- CloudWatch
  - システム監視サービス
    - ✓ 死活監視 / 性能監視 / キャパシティ監視
- CloudWatch Logs
  - ログ管理プラットフォームサービス
    - ✓ EC2上のOS, APPのログ
    - ✓ AWSマネージド サービスのログ
- CloudWatch Events
  - AWS上リソースの状態監視サービス
  - AWSリソースに対するイベントをトリガーにアクションを実行する機能

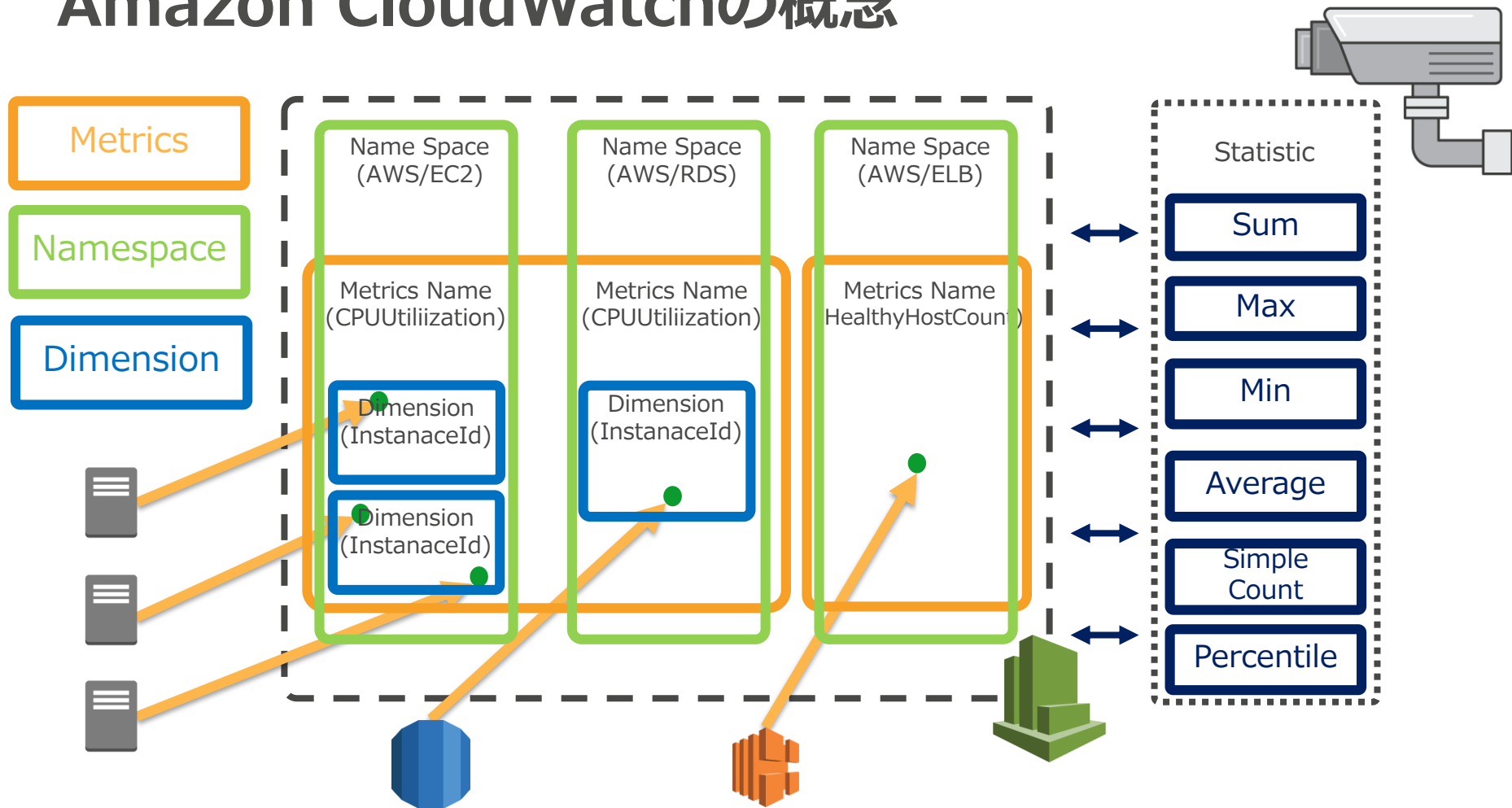
# Amazon CloudWatch



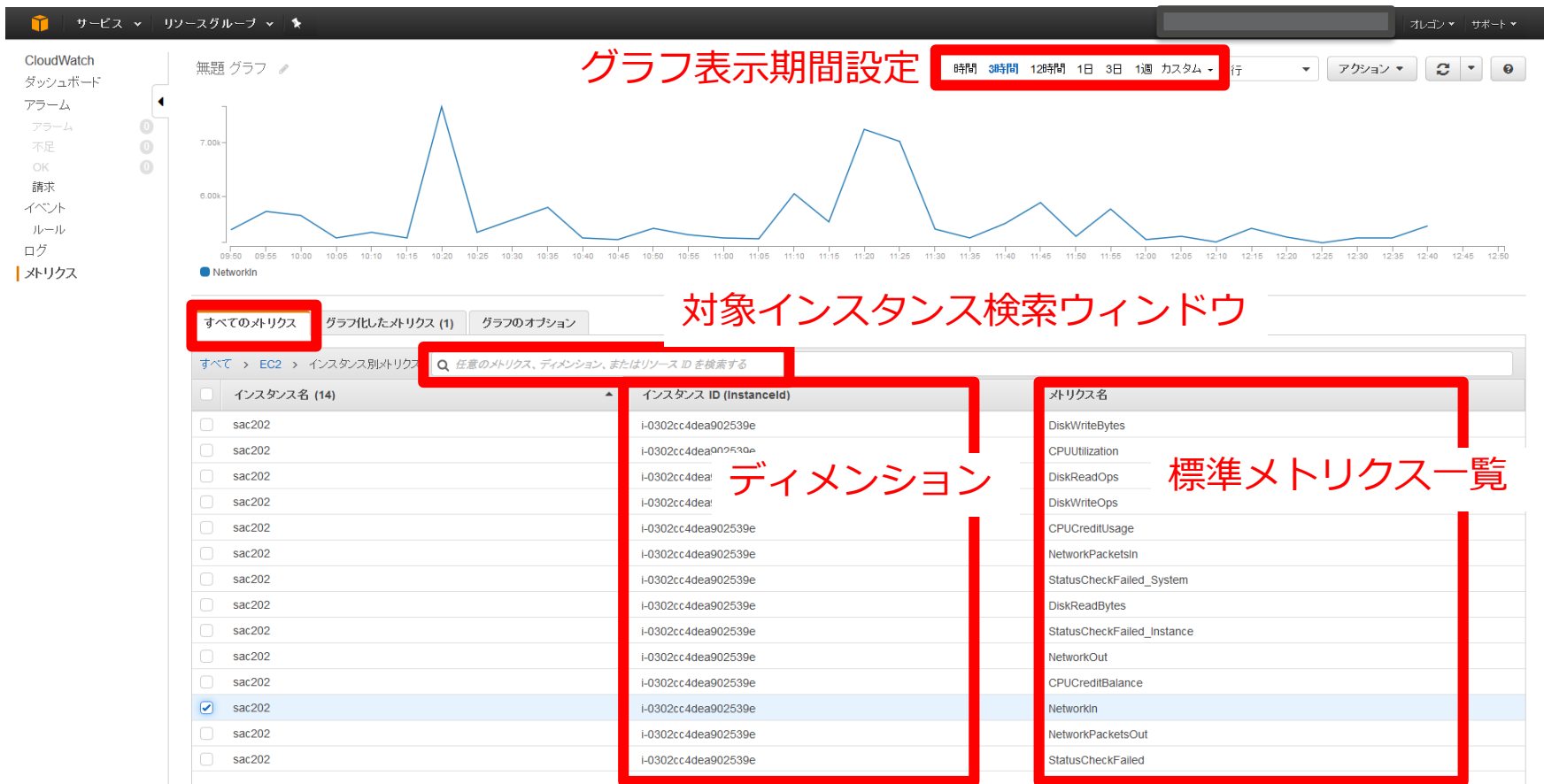
CloudWatch

- CloudWatch
  - システム監視サービス
    - ✓ 死活監視 / 性能監視 / キャパシティ監視
- CloudWatch Logs
  - ログ管理プラットフォーム サービス
    - ✓ EC2上のOS, APPのログ
    - ✓ AWSマネジドサービスのログ
- CloudWatch Events
  - AWS上リソースの状態監視サービス
  - AWSリソースに対するイベントをトリガーにアクションを実行する機能

# Amazon CloudWatchの概念



# CloudWatch利用イメージ 標準メトリックス監視



# CloudWatch利用イメージ 標準メトリックス監視



# CloudWatch のメトリックス値

CloudWatchで取得される情報は**統計情報**

- メトリックスデータを指定した期間で集約したもの
- それぞれのメトリックスについて適切な統計情報を見る必要がある

メトリックスデータの保管は**15ヶ月**まで

- 15ヶ月以上保存する場合は、APIでデータを取得し別の場所に保管しておく（サードパーティ製ツールとの連携も検討）

データの粒度によって遡って参照できる期間が異なる

- 1分毎のデータポイント：15日間
- 5分毎のデータポイント：63日間
- 1時間毎のデータポイント：15ヶ月間

# Amazon CloudWatch を使ったアラーム設定

OK

定義された閾値を  
下回っている  
(正常値)

アラーム  
(Alarm)

定義された閾値を  
上回っている  
(異常値)

不足  
(INSUFFICIENT)

データが不足のため、  
状態を判定できない  
(判定不能)



# Amazon CloudWatch を使ったアラーム設定

OK

定義された閾値を  
下回っている  
(正常値)

アラーム  
(Alarm)

定義された閾値を  
上回っている  
(異常値)

不足  
(INSUFFICIENT)

データが不足のため、  
状態を判定できない  
(判定不能)

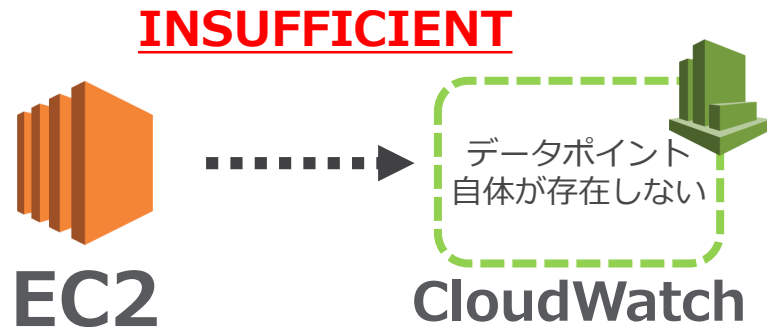
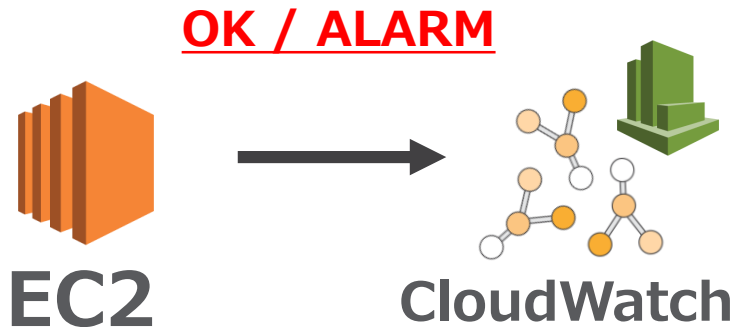
CloudWatch特有のステータス

# INSUFFICIENT\_DATA の考え方

CloudWatchはデータポイントを基準にステータスを判断

- データポイントとはCloudWatchに送信される値(CPU値など)
- OK / アラーム時は入力されたデータポイントを基準に状態評価
- INSUFFICIENT時はCloudWatchにデータポイントの入力が無い状態

→ “INSUFFICIENT”は必ずしも障害を表すステータスではない



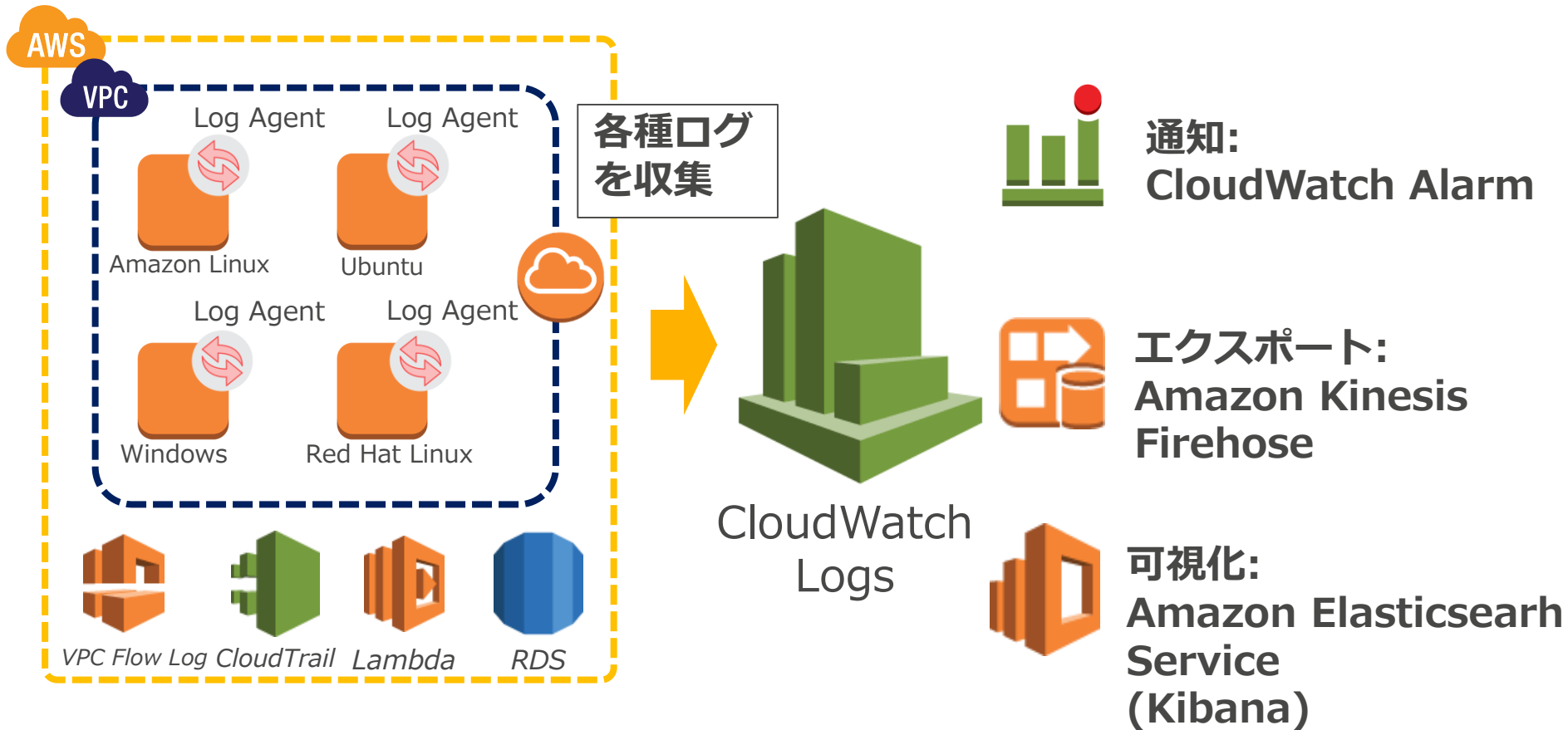
# Amazon CloudWatch Logs



CloudWatch

- CloudWatch
  - AWS上で稼働するシステム監視サービス
    - ✓ 死活監視 / 性能監視 / キャパシティ監視
- CloudWatch Logs
  - **ログ管理**プラットフォームサービス
    - ✓ EC2上のOS, APPのログ
    - ✓ AWSマネジドサービスのログ
- CloudWatch Events
  - AWS上リソースの状態監視サービス
  - AWSリソースに対するイベントをトリガーにアクションを実行する機能

# CloudWatch Logs



# Use cases



- ログの長期保存、ストレージの容量削減
- HTTP responsesのエラー、例外、性能の監視
- ホストにログインせずに障害調査
- セキュリティインシデント対応用の証跡ログとして

# CloudWatch Logsのログ管理

## Log Group



Web Server

## Log Stream



web001.ap-northeast-1



web002.ap-northeast-1



web003.ap-northeast-1

## Log Event



```
2014-07-24 17:20:50 UTC+9 *Jul 24 08:37:00 ip-10-0-10-104 kernel: [ 0.000000] Linux version 3.10.42-1.el6.x86_64 (gcc41@phobos-buil-64003) (~
2014-07-24 17:20:50 UTC+9 *Jul 24 08:37:00 ip-10-0-10-104 kernel: [ 0.000000] Command line: root=LABEL/ console=tty0
2014-07-24 17:20:50 UTC+9 *Jul 24 08:37:00 ip-10-0-10-104 kernel: [ 0.000000] BIOS-provided physical RAM map:
2014-07-24 17:20:50 UTC+9 *Jul 24 08:37:00 ip-10-0-10-104 kernel: [ 0.000000] BIOS-e820: [mem 0x0000000000000000-0x00000000000000ff] usable
2014-07-24 17:20:50 UTC+9 *Jul 24 08:37:00 ip-10-0-10-104 kernel: [ 0.000000] BIOS-e820: [mem 0x0000000000000000-0x00000000000000ff] reserved

2014-07-24 17:20:50 UTC+9 *Jul 24 08:37:00 ip-10-0-10-104 kernel: [ 0.000000] Linux version 3.10.42-1.el6.x86_64 (gcc41@phobos-buil-64003) (~
2014-07-24 17:20:50 UTC+9 *Jul 24 08:37:00 ip-10-0-10-104 kernel: [ 0.000000] Command line: root=LABEL/ console=tty0
2014-07-24 17:20:50 UTC+9 *Jul 24 08:37:00 ip-10-0-10-104 kernel: [ 0.000000] BIOS-provided physical RAM map:
2014-07-24 17:20:50 UTC+9 *Jul 24 08:37:00 ip-10-0-10-104 kernel: [ 0.000000] BIOS-e820: [mem 0x0000000000000000-0x00000000000000ff] usable
2014-07-24 17:20:50 UTC+9 *Jul 24 08:37:00 ip-10-0-10-104 kernel: [ 0.000000] BIOS-e820: [mem 0x0000000000000000-0x00000000000000ff] reserved

2014-07-24 17:20:50 UTC+9 *Jul 24 08:37:00 ip-10-0-10-104 kernel: [ 0.000000] Linux version 3.10.42-1.el6.x86_64 (gcc41@phobos-buil-64003) (~
2014-07-24 17:20:50 UTC+9 *Jul 24 08:37:00 ip-10-0-10-104 kernel: [ 0.000000] Command line: root=LABEL/ console=tty0
2014-07-24 17:20:50 UTC+9 *Jul 24 08:37:00 ip-10-0-10-104 kernel: [ 0.000000] BIOS-provided physical RAM map:
2014-07-24 17:20:50 UTC+9 *Jul 24 08:37:00 ip-10-0-10-104 kernel: [ 0.000000] BIOS-e820: [mem 0x0000000000000000-0x00000000000000ff] usable
2014-07-24 17:20:50 UTC+9 *Jul 24 08:37:00 ip-10-0-10-104 kernel: [ 0.000000] BIOS-e820: [mem 0x0000000000000000-0x00000000000000ff] reserved
```

# ログの保存期間

CloudWatch Logsはログの保存期間を設定可能

## Log Groups

Create Metric Filter

Create Log Group

Delete Log Group

Log Groups		Expire Events After	Metric Filters
<input type="checkbox"/>	CloudTrail-Virginia	Never Expire	1 filter
<input type="checkbox"/>	Linux-Secure-Logs	Never Expire	0 filters
<input type="checkbox"/>	Linux-Sysytem-Logs	Never Expire	1 filter
<input type="checkbox"/>	Windows-Log-Group	Never Expire	0 filters
<input type="checkbox"/>	Windows-SQL-Logs	Never Expire	0 filters

Never Expire ▼

Never Expire  
1 day  
3 days  
5 days  
1 week (7 days)  
2 weeks (14 days)  
1 month (30 days)  
2 months (60 days)  
3 months (90 days)  
4 months (120 days)  
5 months (150 days)  
6 months (180 days)  
1 year (365 days)  
13 months (400 days)  
18 months (545 days)  
2 years (731 days)  
5 years (1827 days)  
10 years (3653 days)

# ログモニタリングイメージ

ログ内容はタイムスタンプとログメッセージ（UTF-8）で構成

The screenshot shows the AWS CloudWatch console interface. The breadcrumb navigation indicates the path: CloudWatch > ロググループ > Default-Log-Group > i-0299505423feee84b. The left sidebar shows the 'CloudWatch' section with 'ダッシュボード' selected. The main area displays a log stream with a filter bar at the top. A red box highlights the filter bar, with the text 'フィルター（検索）' (Filter (Search)) written in red. Below the filter bar, a table of log events is shown. A red box highlights the first five rows of the table, with the text 'タイムスタンプ' (Timestamp) written in red. Another red box highlights the log messages in the same rows, with the text 'ログメッセージ' (Log Message) written in red.

CloudWatch > ロググループ > Default-Log-Group > i-0299505423feee84b

フィルター（検索）

イベントのフィルター

すべて展開 行 テキスト

すべて 30秒 5分 1時間 6時間 1日 1週 カスタム

タイムスタンプ

ログメッセージ

時間 (UTC +00:00)	メッセージ
05-03	[System] [Information] [7036] [Service Control Manager] [WIN-3AM8GUMMNLG.defaultvc-adtest2.local] [The Software Protection service entered the running state.]
14:57:16	[System] [Information] [7036] [Service Control Manager] [WIN-3AM8GUMMNLG.defaultvc-adtest2.local] [The Software Protection service entered the stopped state.]
15:17:32	[System] [Error] [1111] [Microsoft-Windows-TerminalServices-Printers] [WIN-3AM8GUMMNLG.defaultvc-adtest2.local] [Driver Microsoft Print To PDF required for printer Microsoft Print To PDF] [Error] [1111] [Microsoft-Windows-TerminalServices-Printers] [WIN-3AM8GUMMNLG.defaultvc-adtest2.local] [Driver Microsoft Print To PDF required for printer Microsoft Print To PDF]
15:17:33	[System] [Information] [7036] [Service Control Manager] [WIN-3AM8GUMMNLG.defaultvc-adtest2.local] [The Device Setup Manager service entered the running state.]
15:17:35	[System] [Error] [1111] [Microsoft-Windows-TerminalServices-Printers] [WIN-3AM8GUMMNLG.defaultvc-adtest2.local] [Driver Send to Microsoft OneNote 16 Driver required for printer Send to Microsoft OneNote 16] [Error] [1111] [Microsoft-Windows-TerminalServices-Printers] [WIN-3AM8GUMMNLG.defaultvc-adtest2.local] [Driver Send to Microsoft OneNote 16 Driver required for printer Send to Microsoft OneNote 16]
15:19:35	[System] [Information] [7036] [Service Control Manager] [WIN-3AM8GUMMNLG.defaultvc-adtest2.local] [The Device Setup Manager service entered the stopped state.]
15:21:09	[System] [Information] [7036] [Service Control Manager] [WIN-3AM8GUMMNLG.defaultvc-adtest2.local] [The WinHTTP Web Proxy Auto-Discovery Service service entered the stopped state.]
15:34:57	[System] [Error] [36888] [Schannel] [WIN-3AM8GUMMNLG.defaultvc-adtest2.local] [A fatal alert was generated and sent to the remote endpoint. This may result in termination of the connection.]
15:34:57	[System] [Error] [36888] [Schannel] [WIN-3AM8GUMMNLG.defaultvc-adtest2.local] [A fatal alert was generated and sent to the remote endpoint. This may result in termination of the connection.]
15:40:04	[System] [Error] [36888] [Schannel] [WIN-3AM8GUMMNLG.defaultvc-adtest2.local] [A fatal alert was generated and sent to the remote endpoint. This may result in termination of the connection.]
15:40:04	[System] [Error] [36888] [Schannel] [WIN-3AM8GUMMNLG.defaultvc-adtest2.local] [A fatal alert was generated and sent to the remote endpoint. This may result in termination of the connection.]
15:54:42	[System] [Information] [7036] [Service Control Manager] [WIN-3AM8GUMMNLG.defaultvc-adtest2.local] [The WinHTTP Web Proxy Auto-Discovery Service service entered the running state.]



# CloudWatch Logs Metric Filter

## ログイベントから特定の文字列のフィルタリングが可能

### ログメトリクスフィルタの定義

#### ロググループのフィルター: /var/log/messages

メトリクスフィルターを使用し、ロググループ内のイベントが CloudWatch ログに送信されるときに、それらのイベントを自動的にモニタリングできます。特定の用語のモニタリングやカウントを行ったり、ログイベントから値を抽出したりでき、その結果をメトリクスに関連付けることができます。[パターン構文の詳細を確認してください](#)。

#### フィルタパターン

[例の表示](#)

#### テストするログデータの選択

[クリア](#)[パターンのテスト](#)

```
Apr 28 10:22:06 ip-172-31-30-146 dhclient[2133]: XMT: Solicit on eth0, interval 125240ms.
Apr 28 10:24:12 ip-172-31-30-146 dhclient[2133]: XMT: Solicit on eth0, interval 109010ms.
Apr 28 10:24:36 ip-172-31-30-146 dhclient[2047]: DHCPREQUEST on eth0 to 172.31.16.1
Apr 28 10:24:36 ip-172-31-30-146 dhclient[2047]: DHCPACK from 172.31.16.1 (xid=0xb30
Apr 28 10:24:36 ip-172-31-30-146 dhclient[2047]: bound to 172.31.30.146 -- renewal in
```

Metric Filterからアラーム作成、SNS連携が可能

#### 結果

サンプルログの 50 イベントから 3 の一致が見つかりました。

# CloudWatch Logs Metric filter syntax

- 文字列の一致
- Common log format
- JSON



# Metric filters – 文字列の一致

## Filter examples

- Error
- “Invalid user”
- NullPointerException “main(”

## Notes

- AND検索
- 文字列のグループ化、英数字以外の検索は “ ” 囲む
- Metricは出現回数で評価される

# Metric filters – Common log format

## Log examples

```
[11/Nov/2014:02:00:14 +0000] 10.15.128.6 "GET HTTP/1.1" 200 108 33 "S3Console/0.4"  
127.0.0.1 user-identifier frank [10/Oct/2000:13:55:36 -0700] "GET /apache_pb.gif HTTP/1.0"  
200 2326
```

## Filter example

```
[Timestamp, IPAddress, Header, HTTPCode=4*, ...]
```

## Notes

- “” か [ ] で囲まれていない限り、空白文字が区切り文字となる
- 値の抽出が可能

# Metric filters – JSON

## Filter examples

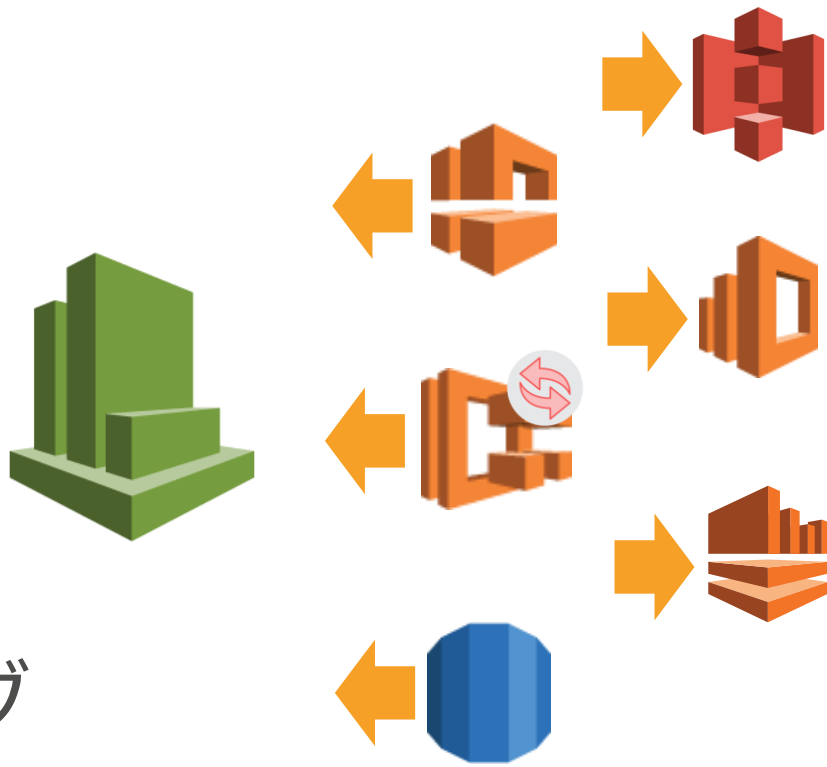
```
{$.errorCode = "AccessDenied" || $.errorCode = "UnauthorizedOperation"}  
{ $.eventType = "UpdateTrail" }  
{ $.sourceIpAdress != 123.123.* }  
{ $.arrayKey[0] = "value" }  
{ $.objectList[1].id = 2 }
```

## Notes

- { } で囲むとJSONとして評価される

# AWSサービスとの連携

- S3へのエクスポート
- VPC Flow Logs
- Elasticsearch
- ECS
- Kinesis
- RDS拡張モニタリング



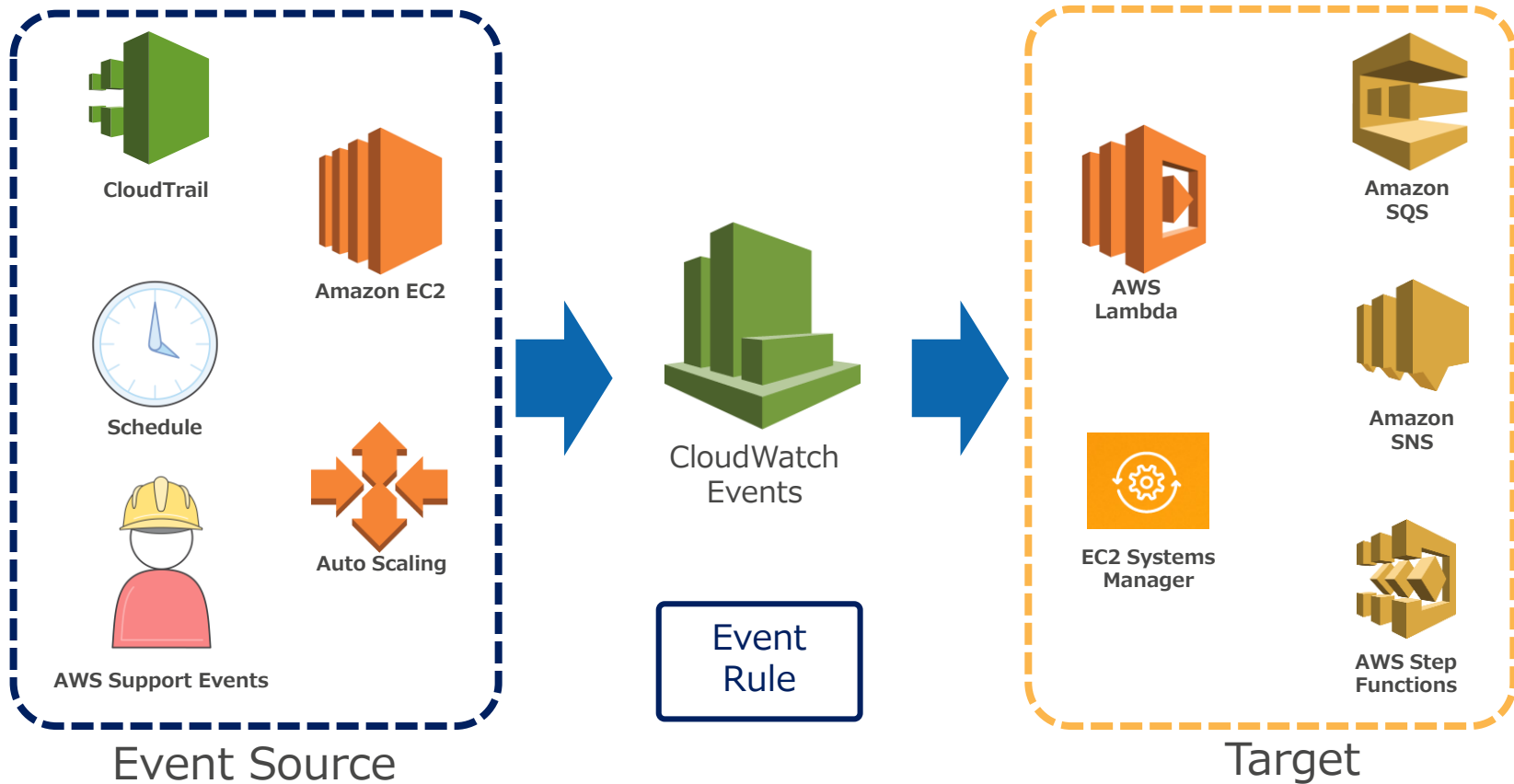
# Amazon CloudWatch Events



CloudWatch

- CloudWatch
  - AWS上で稼働するシステム監視サービス
    - ✓ 死活監視 / 性能監視 / キャパシティ監視
- CloudWatch Logs
  - ログ管理プラットフォーム サービス
    - ✓ EC2上のOS, APPのログ
    - ✓ AWSマネジドサービスのログ
- CloudWatch Events
  - AWS上リソースの状態監視サービス
  - AWSリソースに対する**イベントをトリガーにアクションを実行**する機能

# CloudWatch Events





# イベントソースの選択

## EC2 Instance states change notification

- Pending/Running/Shutting down/Stopped/Stopping/Terminated

## Schedule

- 間隔：分(Minutes)/時間(Hours)/日(Days)
- クーロン表記

## AWS API call

- AWS CloudTrailにより発行されたイベント

## AWS console sign-in

## AWS SupportへのCase作成

## Auto Scaling

- Launch Successful / Launch Unsuccessful  
Terminate Successful / Terminate Unsuccessful

### ステップ 1: ルールの作成

AWS 環境で発生するイベントに基づいてターゲットを呼び出すためのルールを作成します。

#### イベントソース

イベントパターンを構築またはカスタマイズするか、スケジュールを設定してターゲットを呼び出します。

☒ イベントパターン ⓘ ☐ スケジュール ⓘ

サービス別のイベントに一致するイベント パターンの構築 ▼

サービス名

イベントタイプ

☐ 任意の状態 ☒ 特定の状態

☐ 特定のインスタンス

▼ イベントパターンを構築 ▼

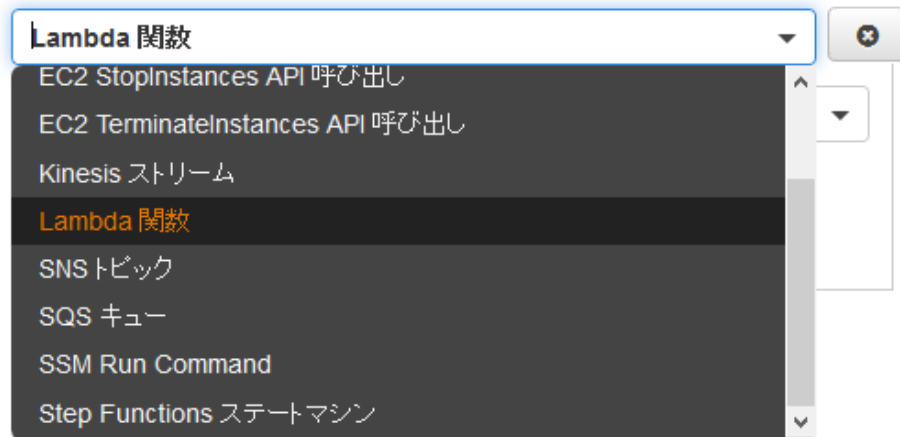
クリップボードにコピー 編集

# ターゲットの選択

- Lambda Function
- SNS Topic
- SQS Queue
- SSM Run Command
- Step Function StateMachine
- Kinesis Stream
- Built-in target
  - EC2インスタンスの再起動
  - EC2インスタンスの削除
  - EC2インスタンスの停止
  - EBSボリュームのスナップショット作成

## ターゲット

イベントがイベントパターンに一致するか、スケジュールがトリガーされたときに呼び出すターゲットを選択します。



# (再々掲) Amazon CloudWatchのできる事



CloudWatch

- CloudWatch
  - システム監視サービス
    - ✓ 死活監視 / 性能監視 / キャパシティ監視
- CloudWatch Logs
  - ログ管理プラットフォームサービス
    - ✓ EC2上のOS, APPのログ
    - ✓ AWSマネージド サービスのログ
- CloudWatch Events
  - AWS上リソースの状態監視サービス
  - AWSリソースに対するイベントをトリガーにアクションを実行する機能

# まとめ



CloudWatch

- クラウドシステムの正常・異常を監視する事ができるマネージドサービス
- サービス間の連携により各種サービス、アプリケーションの運用の自動化が可能
- AWSクラウドにおける運用監視はCloudWatchが最適！

# 本セッションのFeedbackをお願いします

受付でお配りしたアンケートに本セッションの満足度やご感想などをご記入ください  
アンケートをご提出いただきました方には、もれなく**素敵なAWSオリジナルグッズ**を  
プレゼントさせていただきます



アンケートは受付、パミール3FのEXPO展示会場内にて回収させていただきます

AWS

S U M M I T

Thank you!

