

# Amazon CloudWatch / CloudWatch Logs

2015/07/01

AWS Black Belt Tech Webinar 2015

アマゾンデータサービスジャパン株式会社

プロフェッショナル サービス 山本 教仁

パートナー ソリューション アーキテクト 酒徳 知明

# Agenda

- Amazon CloudWatch
- Amazon CloudWatch Logs
- AWSでの運用監視



**Amazon**  
**CloudWatch**

# CloudWatch

[AWS](#)[サービス](#)[EC2](#)[CloudWatch](#)[CloudTrail](#)[Config](#)[S3](#)[編集](#)[東京](#)[サポート](#)

## アマゾン ウェブ サービス

### コンピューティング

[EC2](#)

クラウド内の仮想サーバー

[Lambda](#)

イベント発生時にコードを実行

[EC2 Container Service](#)

Docker コンテナの実行と管理

### ストレージ & コンテンツ配信

[S3](#)

スケーラブルなクラウドストレージ

[Storage Gateway](#)

オンプレミス IT 環境とクラウドストレージの統合

[Glacier](#)

クラウド内のアーカイブストレージ

[CloudFront](#)

グローバルなコンテンツ配信ネットワーク

### データベース

[RDS](#)

マネージド型のリレーショナルデータベースサービス

[DynamoDB](#)

予測可能でスケーラブルな NoSQL データストア

[ElastiCache](#)

インメモリアリキャッシュ

[Redshift](#)

マネージド型のペタバイトスケールのデータウェアハウスサービス

### ネットワーキング

### 管理およびセキュリティ

[Directory Service](#)

クラウド上の管理型ディレクトリ

[Identity & Access Management](#)

アクセスコントロールとキー管理

[Trusted Advisor](#)

AWS クラウド最適化エキスパート

[CloudTrail](#)

ユーザーアクティビティと変更の追跡

[Config](#)

リソース構成および変更の追跡

[CloudWatch](#)

リソースとアプリケーションのモニタリング

### デプロイ & マネジメント

[Elastic Beanstalk](#)

AWS アプリケーションコンテナ

[OpsWorks](#)

DevOps アプリケーション管理サービス

[CloudFormation](#)

テンプレートによる AWS リソース作成

[CodeDeploy](#)

自動デプロイ

### 分析

[Elastic MapReduce](#)

マネージド型 Hadoop フレームワーク

[Kinesis](#)

ビッグデータストリームのリアルタイム処理

[Data Pipeline](#)

### アプリケーションサービス

[SQS](#)

メッセージキューサービス

[SWF](#)

アプリケーションコンポーネントを連携させるワークフローサービス

[AppStream](#)

低レイテンシーのアプリケーションストリーミング

[Elastic Transcoder](#)

使いやすいスケーラブルなメディア変換サービス

[SES](#)

E メール送信サービス

[CloudSearch](#)

マネージド型検索サービス

### モバイルサービス

[Cognito](#)

ユーザー ID およびアプリケーションデータの同期

[Mobile Analytics](#)

大規模なアプリケーションの使用状況データの把握

[SNS](#)

プッシュ通知サービス

### エンタープライズアプリケーション

[WorkSpaces](#)

クラウド内のデスクトップ

[WorkDocs](#)

セキュアなエンタープライズ向けストレージおよび共有サービス

[WorkMail](#) プレビュー

サードパーティ提供された E メール、 calendaring、および

## リソースグループ

リソースグループは、1 つ以上のタグを共有するリソースのコレクションです。お客様のアカウントの各プロジェクトのグループ、アプリケーション、環境の作成

[グループの作成](#)[タグエディター](#)

## その他のリソース

### はじめに

サービスを初めて使用する手順やさらに詳しい使用方法については、ドキュメントを参照してください。

### AWS Console モバイルアプリ

Amazon アプリストア、Google Play、または iTunes から入手可能な AWS コンソールモバイルアプリを使用して、先方でリソースを表示します。

### AWS Marketplace

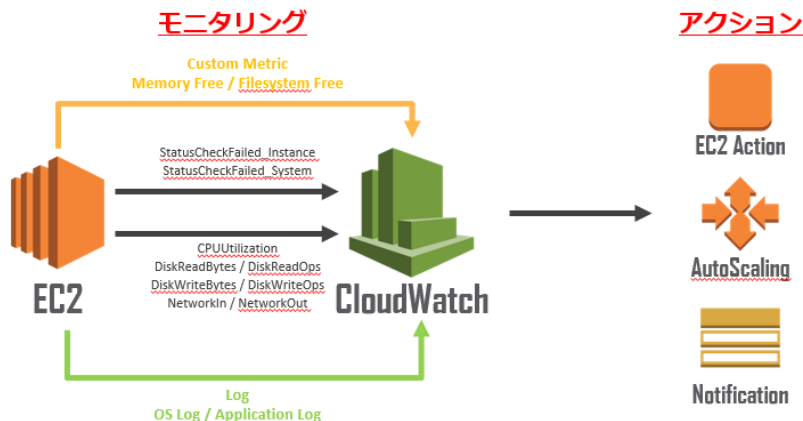
ソフトウェアを検索して購入し、1-Click で起動し、時間単位で料金を支払えます。

### AWS Summit – サンフランシスコ

詳細については、サンフランシスコで開催される AWS Summit で発表予定のエキサイティングな新機能やサービス

# Amazon CloudWatchとは

- AWSの各種リソースを監視するサービス
  - AWSリソースの死活、性能、ログ監視 (監視)
  - 取得メトリックスのグラフ化 (可視化)
  - 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



# CloudWatchに対応するAWSサービス

AWS サービス	名前空間
Auto Scaling	AWS/AutoScaling
AWS Billing	AWS/Billing
Amazon CloudFront	AWS/CloudFront
Amazon CloudSearch	AWS/CloudSearch
Amazon DynamoDB	AWS/DynamoDB
Amazon ElastiCache	AWS/ElastiCache
Amazon Elastic Block Store	AWS/EBS
Amazon Elastic Compute Cloud	AWS/EC2
Elastic Load Balancing	AWS/ELB
Amazon Elastic MapReduce	AWS/ElasticMapReduce
Amazon Kinesis	AWS/Kinesis
Amazon Machine Learning	AWS/ML
AWS OpsWorks	AWS/OpsWorks
Amazon Redshift	AWS/Redshift

AWS の製品	名前空間
Amazon Relational Database Service	AWS/RDS
Amazon Route 53	AWS/Route53
Amazon Simple Notification Service	AWS/SNS
Amazon Simple Queue Service	AWS/SQS
Amazon Simple Workflow Service	AWS/SWF
AWS Storage Gateway	AWS/StorageGateway
Amazon WorkSpaces	AWS/WorkSpaces

[http://docs.aws.amazon.com/ja\\_jp/AmazonCloudWatch/latest/DeveloperGuide/supported\\_services.html](http://docs.aws.amazon.com/ja_jp/AmazonCloudWatch/latest/DeveloperGuide/supported_services.html)

# CloudWatchでの監視データ管理

Metrics

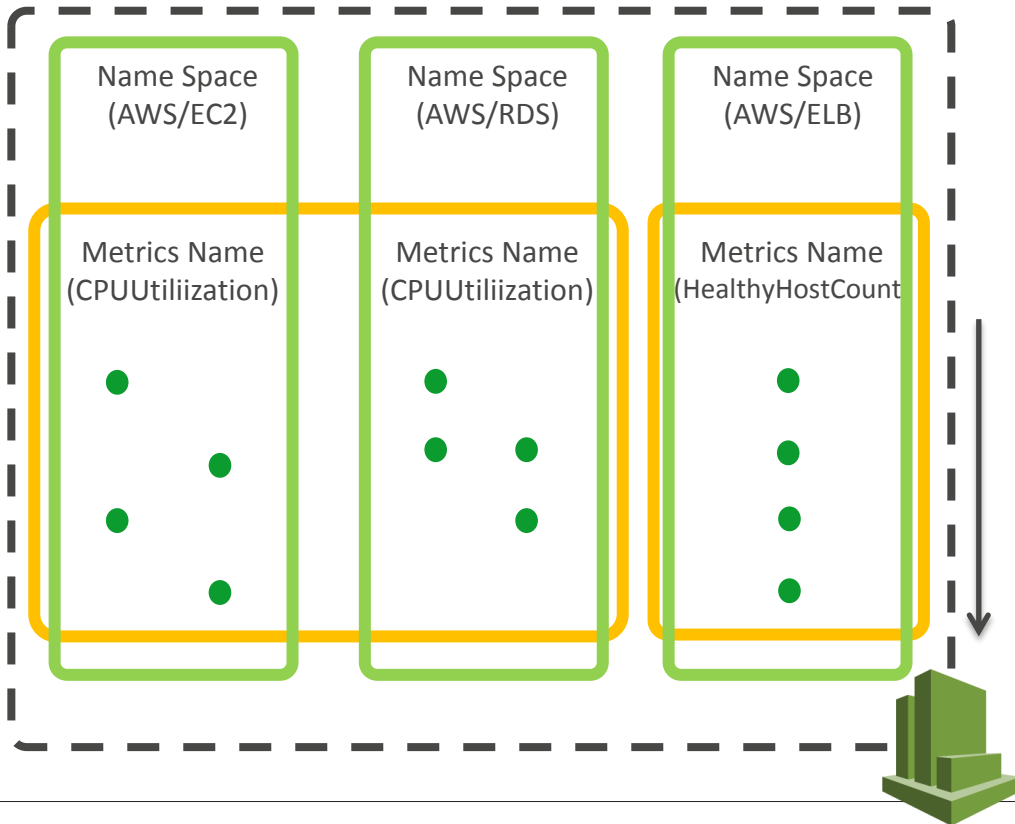
Namespace

## Metrics

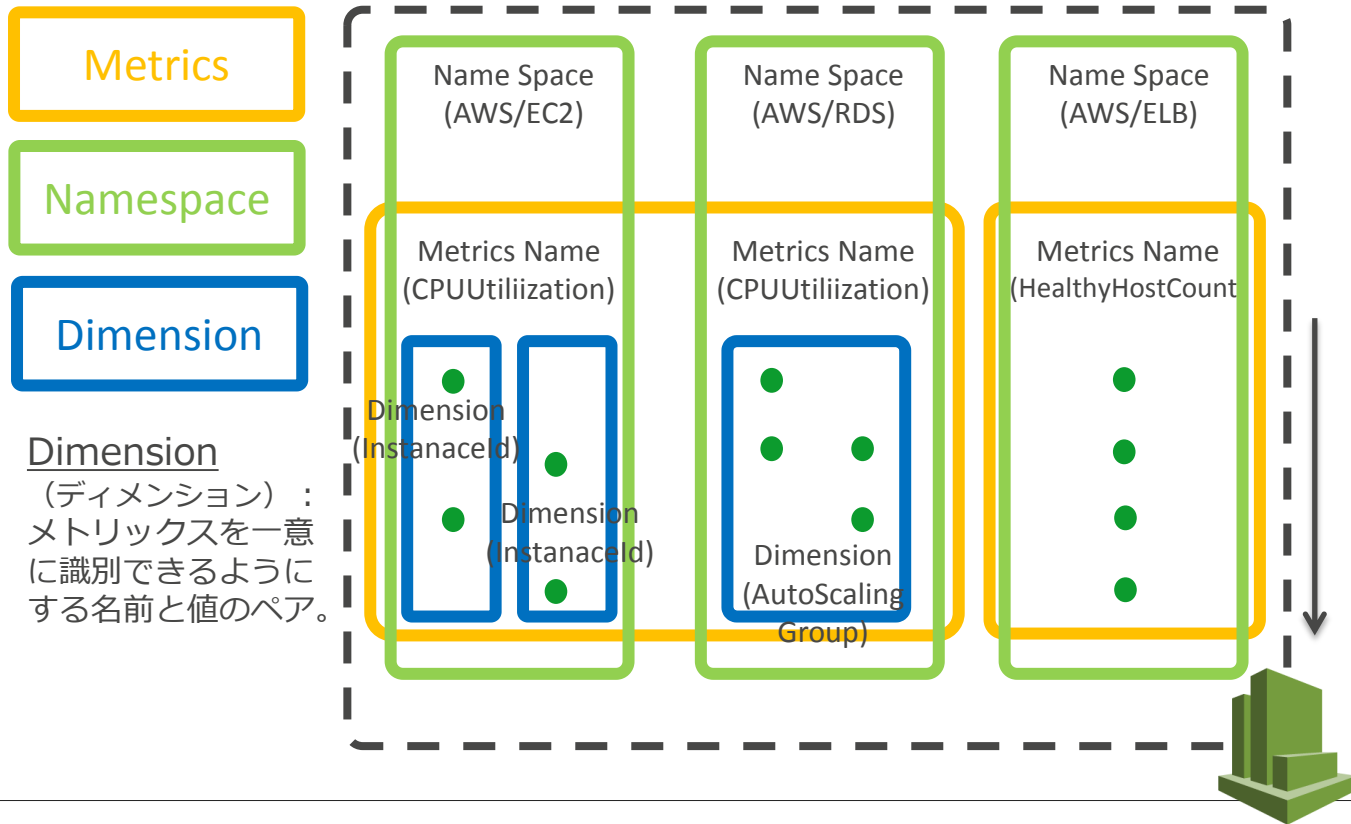
(メトリックス) :  
時系列で表わした  
データポイント一式

## Namespace

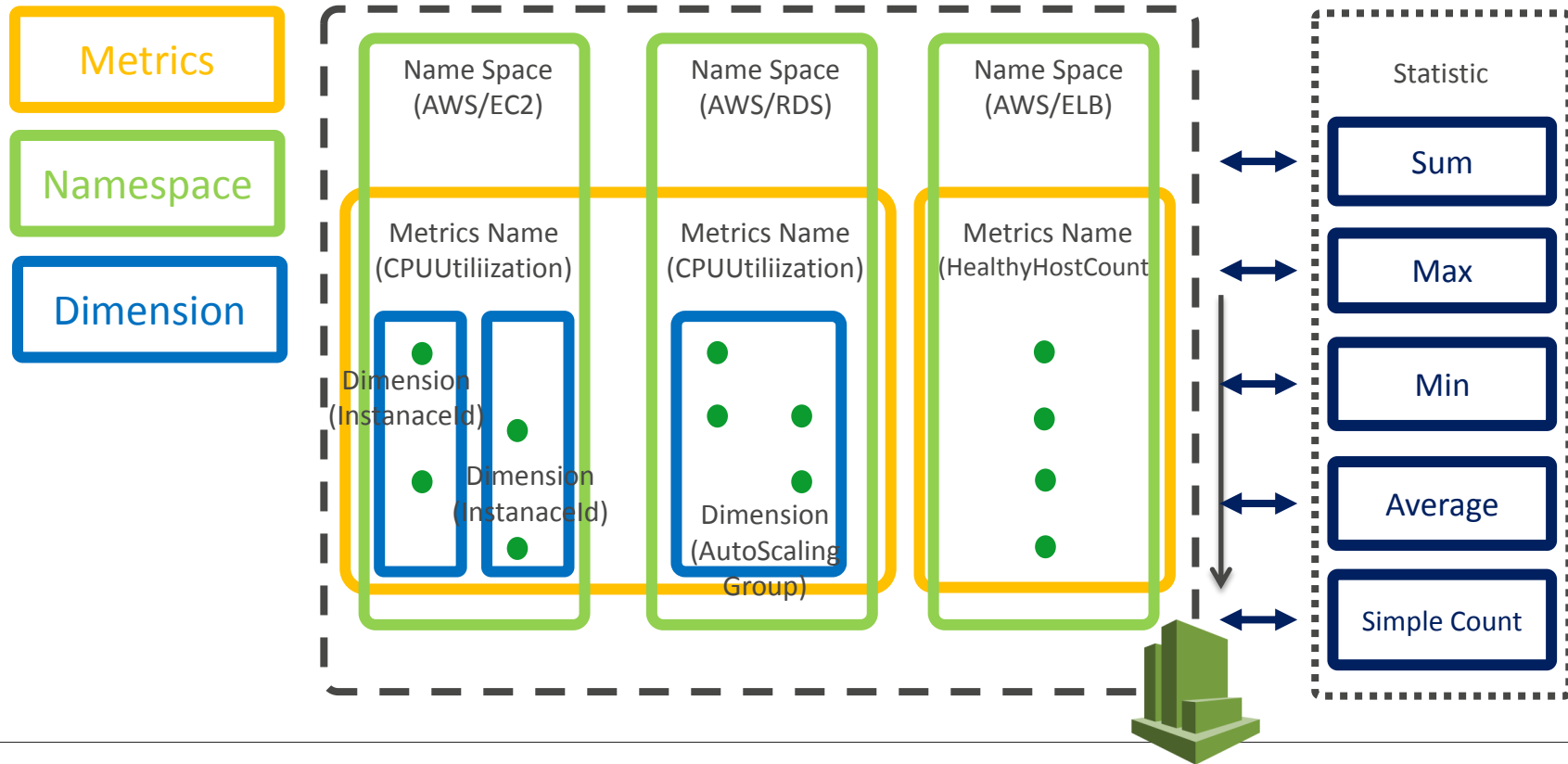
(名前空間) :  
メトリックスの  
コンテナ。標準では  
AWSサービスごと。



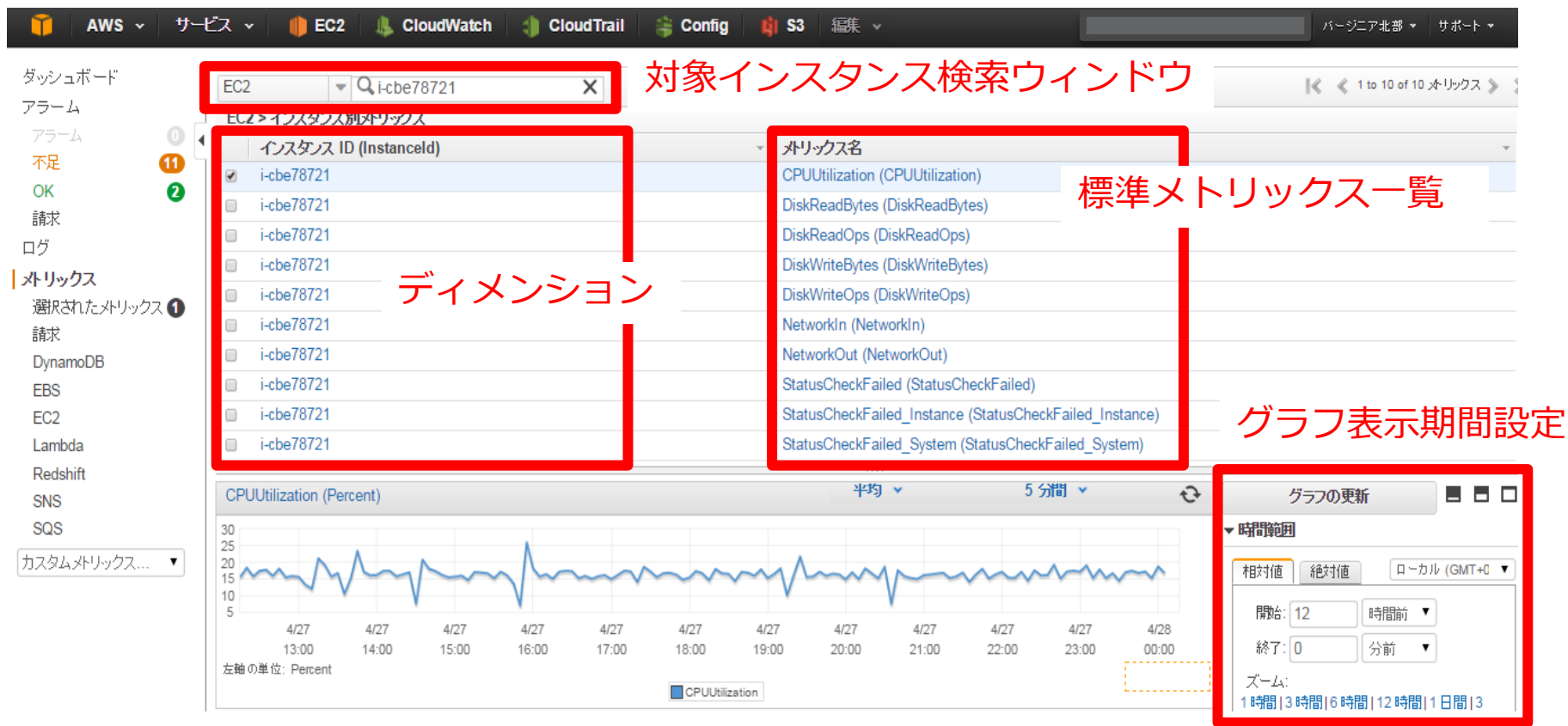
# CloudWatchでの監視データ管理



# CloudWatchでの監視データ管理



# 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

## 標準メトリックス(EC2)

CPUUtilization  
CPUCreditBalance  
CPUCreditUsage  
DiskReadBytes  
DiskWriteBytes  
DiskWriteOps  
NetworkOut  
NetworkIn  
StatusCheckFailed\_Instance  
StatusCheckFailed  
StatusCheckFailed\_System

## 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



# CloudWatch カスタムメトリックス

- 標準メトリックス以外の独自メトリックスも監視可能
  - AWS CLIの"put-metric-data"、API Toolsの"mon-put-data"、もしくは"PutMetricData" APIでデータを登録
  - サイズ制限として、HTTP GETは8KB、HTTP POSTは40KB、1つのPutMetricDataリクエストに20データ

```
$ aws cloudwatch put-metric-data --metric-name RequestLatency\  
  --namespace "GetStarted"\  
  --timestamp 2014-10-28T12:30:00Z\  
  --value 87 \  
  --unit Milliseconds\
```

←単一値の登録

```
$ aws cloudwatch put-metric-data --metric-name RequestLatency¥  
  --namespace "GetStarted"¥  
  --timestamp 2014-10-28T12:30:00Z\  
  --statistic-value Sum=60,Minimum=15,Maximum=105,SampleCount=5
```

←統計セットの登録

- APIコールにスロットリングあり
  - カスタムメトリックスの頻繁な登録や頻度の高いデータ取得には注意

# 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

統計	説明
Minimum	指定された期間に認められた最小値です。この値を用いて、アプリケーションの低ボリュームのアクティビティを判断できます。
Maximum	指定された期間に認められた最大値です。この値を用いて、アプリケーションの高ボリュームのアクティビティを判断できます。
Sum	該当するメトリックスで加算されたすべての合計値です。この統計は、メトリックスの合計ボリュームを判断するのに役立ちます。
Average	指定した期間の $\text{Sum}/\text{SampleCount}$ の値です。この統計を <b>Minimum</b> および <b>Maximum</b> と比較することで、メトリックスの全容、および平均使用量がどれくらい <b>Minimum</b> と <b>Maximum</b> に近いかを判断できます。この比較は、必要に応じてリソースを増減させるべきかを知るのに役立ちます。
SampleCount	統計計算で使用するデータポイントのカウント(数)です。

- メトリックスデータの保管は2週間まで
  - 2週間以上保存する場合は、get-metric-statisticsでデータを取得し別の場所に保管しておく
- データ保管粒度は最短で1分間隔
  - 多くのサービスで1分間隔、5分間隔のものもある

[http://docs.aws.amazon.com/ja\\_jp/AmazonCloudWatch/latest/DeveloperGuide/cloudwatch\\_concepts.html](http://docs.aws.amazon.com/ja_jp/AmazonCloudWatch/latest/DeveloperGuide/cloudwatch_concepts.html)

# Amazon EC2のモニタリングタイプ

## 基本モニタリング

**無料**

**データは  
5分間隔のものを  
閲覧可能**

## 詳細モニタリング

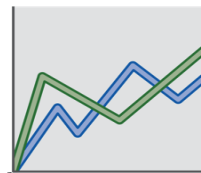
**追加料金が必要**

**データは  
1分間隔のものを  
閲覧可能**

# CloudWatchを使ったマネージドサービスの監視

## • AWSマネージドサービスの監視

- サードパーティツールのエージェントをインストールできないため、CloudWatchでの監視が必須



ELB

- Latency
- BackendConnectionErrors
- HealthyHostCount
- UnHealthyHostCount
- RequestCount
- HTTPCode\_ELB\_5XX
- HTTPCode\_Backend\_4XX



Amazon  
RDS

- CPUUtilization
- FreeableMemory
- SwapUsage
- FreeStorageSpace
- DiskQueueDepth
- ReadIOPS
- ReadThroughput
- ReadLatency
- NetworkReceiveThroughput
- NetworkTransmitThroughput
- WriteIOPS
- WriteThroughput
- WriteLatency
- DatabaseConnections
- BinLogDiskUsage

[http://docs.aws.amazon.com/ja\\_jp/ElasticLoadBalancing/latest/DeveloperGuide/US\\_MonitoringLoadBalancerWithCW.html](http://docs.aws.amazon.com/ja_jp/ElasticLoadBalancing/latest/DeveloperGuide/US_MonitoringLoadBalancerWithCW.html)

[http://docs.aws.amazon.com/ja\\_jp/AmazonCloudWatch/latest/DeveloperGuide/rds-metricscollected.html](http://docs.aws.amazon.com/ja_jp/AmazonCloudWatch/latest/DeveloperGuide/rds-metricscollected.html)

# 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)

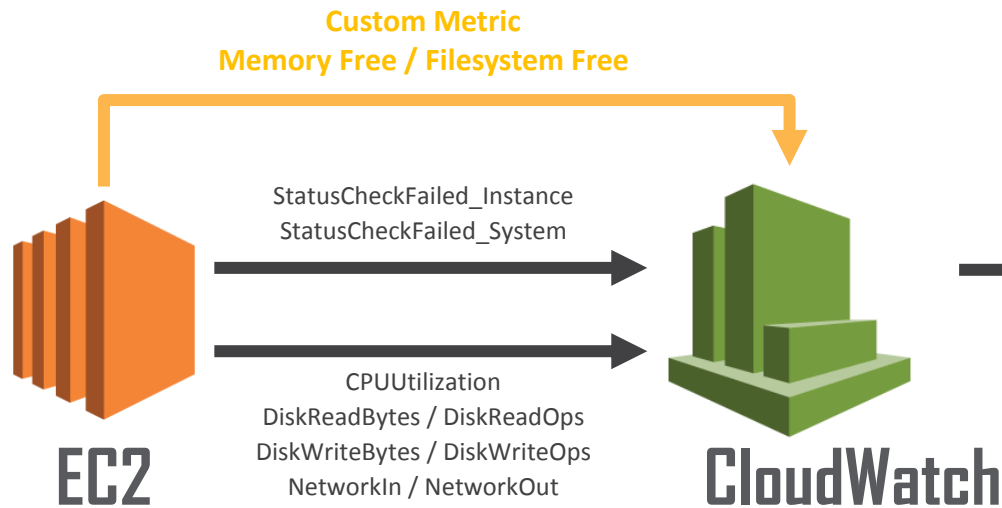
定義された閾値を  
上回っている  
(異常値)

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のアクション機能

## モニタリング



## アクション



Notification



EC2 Action

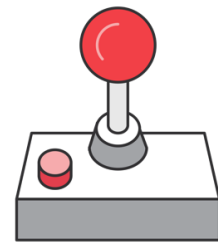


AutoScaling

# CloudWatchのアクション機能

- 各アラーム状態に対してアクションを定義可能
  - 通知 (Notification)
    - Amazon Simple Notification Service (SNS) を使って通知
    - メール送信やHTTP(S)送信、Amazon Simple Queue Service (SQS) への送信が可能
  - EC2アクション
    - EC2インスタンスの復元、停止および終了が実行可能
  - Auto Scalingアクション
    - Auto Scaling GroupのScaling Policyを指定し、インスタンスのスケールアウト／インが可能

# Amazon EC2 Auto Recovery



- EC2の自動復旧
  - EC2インスタンスが稼働しているAWSシステムに障害が発生した場合に、自動的にEC2インスタンス復旧する機能。
    - ネットワーク接続喪失
    - システム電源喪失
    - 物理ホストの障害
- 対応するインスタンスタイプ
  - C3, C4, M3, R3, T2インスタンス
- VPC内のインスタンス
  - EC2クラシックは未対応
  - ハードウェア専有インスタンスは未対応
- EBS-Backedインスタンスのみ



[http://docs.aws.amazon.com/ja\\_jp/AWSEC2/latest/UserGuide/ec2-instance-recover.html](http://docs.aws.amazon.com/ja_jp/AWSEC2/latest/UserGuide/ec2-instance-recover.html)

# CloudWatchアラームの設定

EC2 ダッシュボード

イベント

タグ

レポート

制限

インスタンス

インスタンス

スポットリクエスト

リザーブドインスタンス

イメージ

AMI

バンドルタスク

ELASTIC BLOCK STORE

ボリューム

スナップショット

ネットワーク & セキュリティ

セキュリティグループ

Elastic IP

プレースメントグループ

インスタンスの作成

接続

アクション

タグや属性によるフィルタ, またはキーワードによる検索

<input type="checkbox"/>	Name	インスタンス ID	インスタンスタイプ	アベイラビリティゾーン	インスタンスの状態	ステータスチェック	パブリック IP	プライベート IP アドレス
<input checked="" type="checkbox"/>	cwl-sqlserver	i-cbe78721	m3.medium	us-east-1c	running	2/2 のチェックに合格しました	54.172.19.212	10.0.20.77
<input type="checkbox"/>	Testec2-ami-linux	i-156c2e22	t2.micro	us-east-1b	running	2/2 のチェックに合格しました	52.1.19.37	172.31.63.135
<input type="checkbox"/>	CloudTrail CWL	i-156c2e22	t2.micro	us-east-1c	running	2/2 のチェックに合格しました	54.88.190.17	10.0.1.104
<input type="checkbox"/>	web-id-001	i-156c2e22	t2.micro	us-east-1b	running	2/2 のチェックに合格しました	52.4.74.227	172.31.63.209

インスタンス: **i-cbe78721 (cwl-sqlserver)** Elastic IP: 54.172.19.212

説明

ステータスチェック

モニタリング

タグ

CloudWatch アラーム: OK の 1

CloudWatch メトリクス: 基本モニタリング. [詳細モニタリングを有効化](#)

次のデータを表示: 過去 1 時間

以下は、選択されたリソースの CloudWatch メトリクスです (最大 10)。画面を拡大するには、グラフをクリックします。すべての時刻は協定世界時 (UTC) で表示されています。 [すべての CloudWatch メトリクスを表示](#)

CPU 使用率 (%) (パーセント)

ディスク読み取り (Bytes)

ディスク読み取り操作 (操作)

ディスク書き込み (Bytes)

アラームの作成

amazon  
web services

20

# CloudWatchアラームの設定

AWS サービス EC2 CloudWatch CloudTrail Config S3 編集

EC2 ダッシュボード  
イベント  
タグ  
レポート  
制限

インスタンス  
インスタンス  
スポットリクエスト  
リザーブドインスタンス

イメージ  
AMI  
バンドルタスク

ELASTIC BLOCK STORE  
ボリューム  
スナップショット

ネットワーク & セキュリティ  
セキュリティグループ  
Elastic IP  
プレイングメントグループ

## アラームの作成

CloudWatch アラームを使用すると、メトリクスデータがお客様の設定したレベルに達したときに、自動的に通知されます。  
アラームを編集するには、まず通知先を選択してから、通知を送信するタイミングを設定します。

☒ 通知の送信先: 手動でトピック名を入力... [キャンセル](#)  
受信者: awsAccount@domain.com

☐ アクションを実行:  
● このインスタンスを復元する ⓘ  
● このインスタンスを停止する ⓘ  
● このインスタンスを終了する ⓘ

次の時: 平均 / CPU 使用率(%)  
状況: >= 80 パーセント  
最低発生数: 3 度次の間隔で発生 5 分

CPU 使用率(%) パーセント

アラームの作成

- CPU使用率を監視対象
- CPU使用率80%以上が3期間（ここでは1期間=5分）以上

アラームの作成

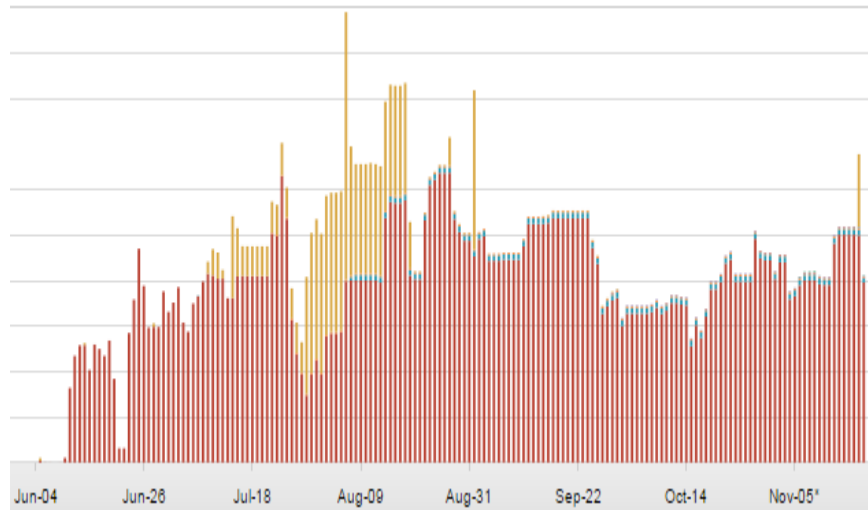
CPU 使用率(%) (パーセント) ディスク読み取り (Bytes) ディスク読み取り操作 (操作) ディスク書き込み (Bytes)

# CloudWatchによるコストの監視

## • Billingアラーム設定

※Virginiaリージョンから設定

- 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



### 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

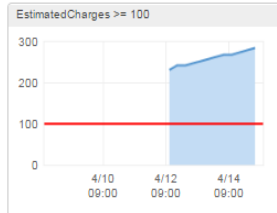
+ Notification

+ AutoScaling Action

+ EC2 Action

### Alarm Preview

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



Namespace: AWS/Billing

Currency: USD

Metric Name: EstimatedCharges

# CloudWatchの料金

- 初期費用無しの従量課金
- 標準の監視は無料
  - EC2インスタンスの標準監視（5分間隔）
  - EBS、ELB、RDSは1分間隔が無料
- アラームやカスタムメトリックスは一定数まで無料
  - 10メトリックス、10 アラーム、および100万APIリクエスト
  - 1 か月あたり5GBのデータの取り込みおよび5GBのアーカイブされたストレージ
- 課金対象及び料金（2015年7月現在 Tokyoリージョン）
  - EC2詳細モニタリング1インスタンスにつき\$3.50/月
  - カスタムメトリックス1つにつき\$0.50/月
  - アラーム1つにつき\$0.10/月
  - APIリクエスト1000回につき\$0.01（Get, List, Putごとに）

<http://aws.amazon.com/jp/cloudwatch/pricing/>

# メンテナンスイベントの監視

- EC2やRDSのメンテナンスイベント（※）

- AWSが予定する、再起動、停止/開始、またはリタイアなどのイベント
  - [http://docs.aws.amazon.com/ja\\_jp/AWSEC2/latest/UserGuide/monitoring-instances-status-check\\_sched.html](http://docs.aws.amazon.com/ja_jp/AWSEC2/latest/UserGuide/monitoring-instances-status-check_sched.html)
- メンテナンスイベントが予定されている場合メールで通知されますが、見逃さないようにコマンドラインでの確認をしておくことが有効
- 管理コンソールからも確認可能

## ■ コマンドラインによるイベントの確認例

```
$ aws ec2 describe-instance-status \
    --query "InstanceStatuses[?Events != null].[InstanceId,Events]"

[
  [
    "i-xxxxxxx",
    [
      {
        "Code": "instance-stop",
        "Description": "The instance is running on degraded hardware",
        "NotBefore": "2015-07-01T00:00:00.000Z"
      }
    ]
  ]
]
```

## ■ 管理コンソールでの確認

The screenshot shows the AWS Management Console interface. At the top, there's a search bar and filters. Below, a table lists EC2 instances. One instance, 'Server 2' (ID: i-xxxxxxx), is highlighted. Its status is 'running'. A yellow warning banner indicates a scheduled maintenance event: 'リタイア: このインスタンスは 2015年7月01日 9:00:00 UTC+9 後にリタイアが予定されています。' (Retirement: This instance is scheduled for retirement on 2015-07-01 at 9:00:00 UTC+9). Below the banner, the '説明' (Description) tab is selected, showing details about the instance's status, type, and DNS. A red arrow points from the event details in the banner to the instance's status bar.

インスタンス ID	パブリック DNS
i-xxxxxxx	ec2-xx-xx-xx-xx.ap-northeast-1.compute.amazonaws.com

インスタンスの状態: running  
インスタンスタイプ: t2.micro  
プライベート DNS: ip-xxx-xxx-xx-xx.ap-northeast-1.compute.internal  
パブリック IP: Elastic IP  
アベイラビリティゾーン: ap-northeast-1-c

セキュリティグループ: 既定のセキュリティグループ  
予定されているイベント: **予定されているイベントが 1 件あります**  
AMI ID: amazon-ami-hvm-2015.03.0.x86\_64-gp2

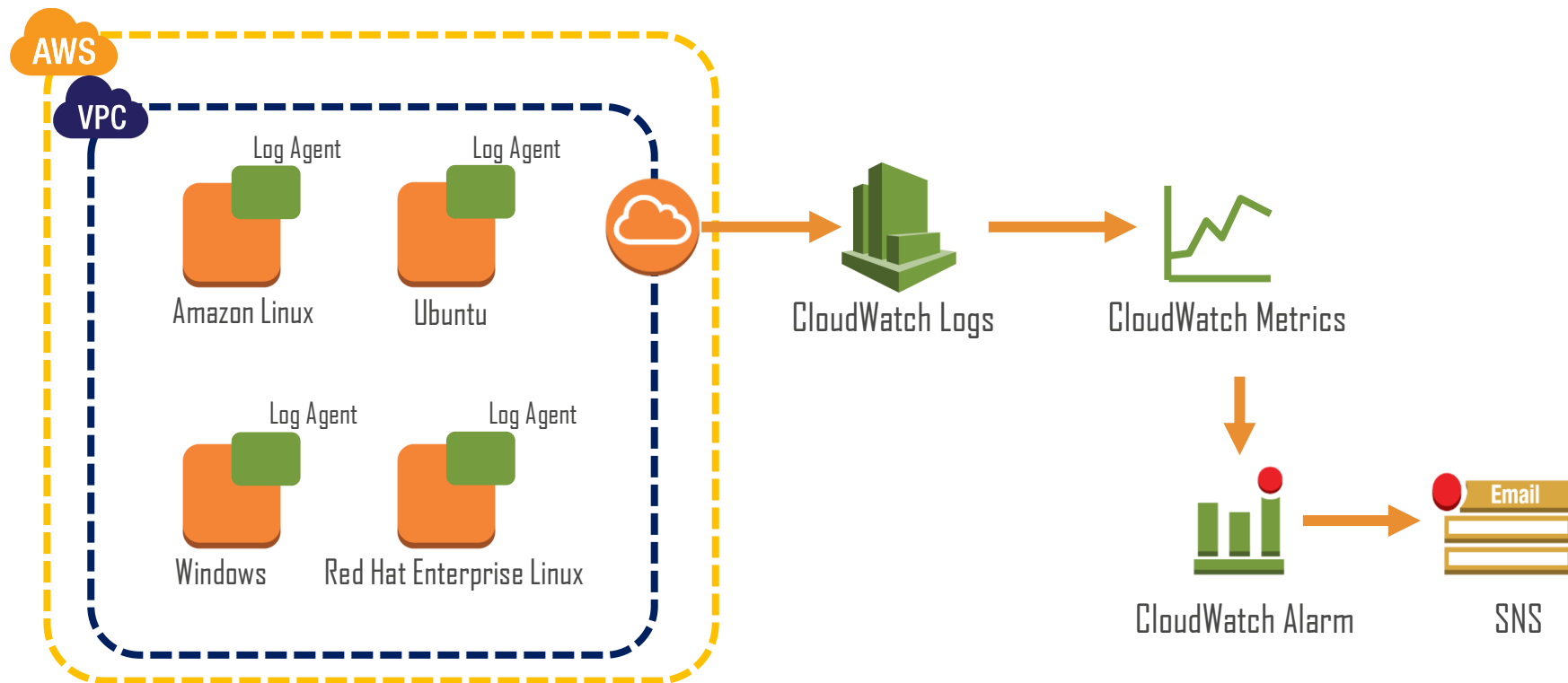


# Amazon CloudWatch Logs

# CloudWatchを使ったログ監視

- CloudWatch Logsを使ったログ監視
- OS、アプリケーション等のテキストログをモニタリング
- エージェント経由でログメッセージをCloudWatchエンドポイントに転送
- ログデータの保存期間は設定可能
  - 1日～永久保存で選択可能

# CloudWatch Logsの利用イメージ



# CloudWatch Logsのディレクトリ階層

## Log Group



Web Server

## Log Stream



web001.ap-northeast-1



web002.ap-northeast-1



web003.ap-northeast-1

## Log Event



# 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 displays the AWS CloudWatch console interface. At the top, there's a navigation bar with various AWS services like AWS, サービス, EC2, CloudWatch, CloudTrail, Config, S3, and 編集. Below this, the left sidebar contains navigation links: ダッシュボード, アラーム, アラーム, 不足, OK, 請求, ログ, メトリクス, 選択されたメトリクス, 請求, DynamoDB, EBS, EC2, Lambda, Redshift, SNS, SQS, and カスタムメトリクス. The main content area shows the 'ロググループ > Windows-Log-Group のストリーム > i-cbe78721 のイベント' view. It includes a filter bar with '日付/時刻' (2015/04/27 06:02:37) and 'ローカル (GMT+09:00)'. Below this is a table of log events, with the 'イベントデータ' column highlighted by a red box. The events are listed with their timestamps and messages, such as 'Windows Modules Installer サービスは 実行中 状態に移行しました。' and 'Windows Modules Installer サービスは 停止 状態に移行しました。'.

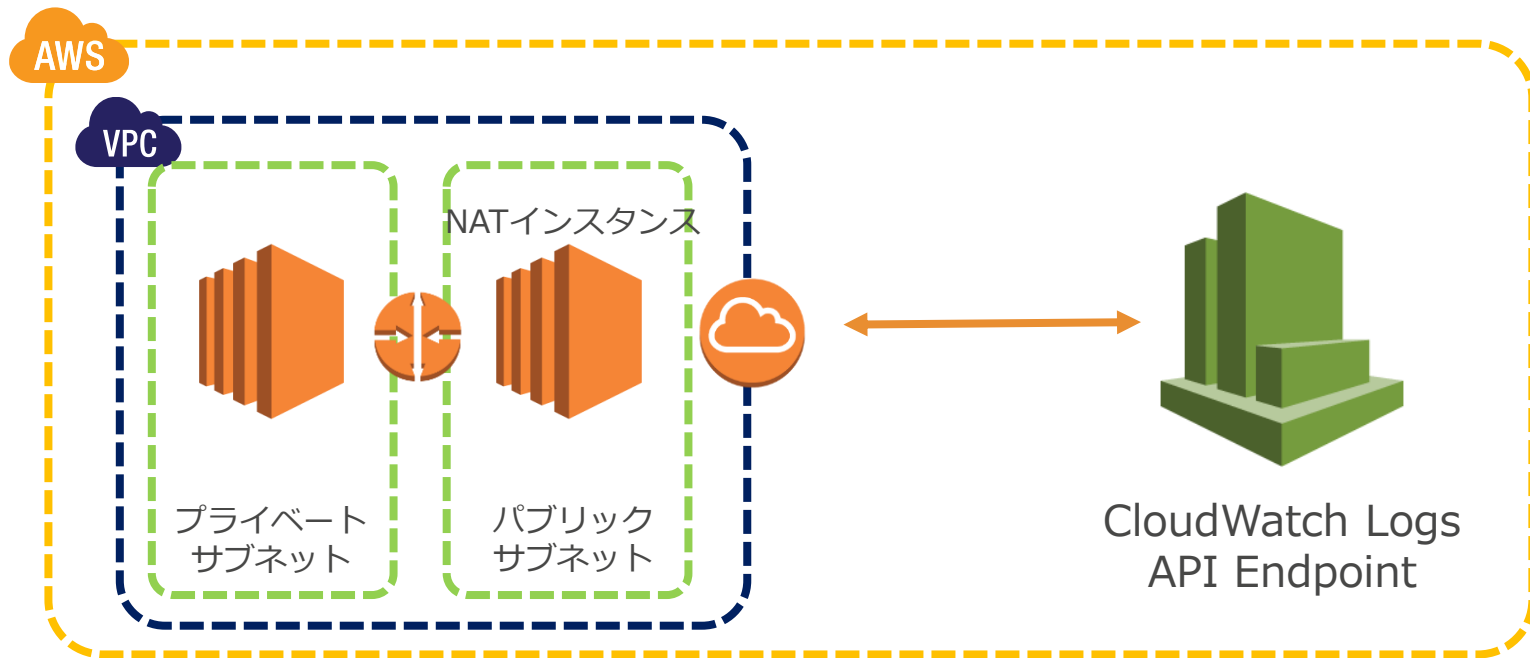
日付/時刻	作成時刻	イベントデータ
2015-04-27 06:02:37 UTC+9	▶ [System] [Information] [7036] [Service Control Manager] [WIN-BHGRATORDMN] [Windows Modules Installer サービスは 実行中 状態に移行しました。]	
2015-04-27 06:04:42 UTC+9	▶ [System] [Information] [7036] [Service Control Manager] [WIN-BHGRATORDMN] [Windows Modules Installer サービスは 停止 状態に移行しました。]	
2015-04-27 06:04:42 UTC+9	▶ [System] [Information] [7036] [Service Control Manager] [WIN-BHGRATORDMN] [Windows Modules Installer サービスは 停止 状態に移行しました。]	
2015-04-27 06:06:35 UTC+9	▶ [System] [Information] [7036] [Service Control Manager] [WIN-BHGRATORDMN] [Windows Modules Installer サービスは 実行中 状態に移行しました。]	
2015-04-27 06:06:35 UTC+9	▶ [System] [Information] [7036] [Service Control Manager] [WIN-BHGRATORDMN] [Windows Modules Installer サービスは 実行中 状態に移行しました。]	
2015-04-27 06:10:05 UTC+9	▶ [System] [Information] [7036] [Service Control Manager] [WIN-BHGRATORDMN] [Windows Modules Installer サービスは 停止 状態に移行しました。]	
2015-04-27 06:10:05 UTC+9	▶ [System] [Information] [7036] [Service Control Manager] [WIN-BHGRATORDMN] [Windows Modules Installer サービスは 停止 状態に移行しました。]	
2015-04-27 06:38:28 UTC+9	▶ [System] [Information] [7036] [Service Control Manager] [WIN-BHGRATORDMN] [WinHTTP Web Proxy Auto-Discovery Service サービスは 停止 状態に移行しまし…]	
2015-04-27 06:57:57 UTC+9	▶ [System] [Information] [7036] [Service Control Manager] [WIN-BHGRATORDMN] [WinHTTP Web Proxy Auto-Discovery Service サービスは 実行中 状態に移行しま…]	
2015-04-27 07:25:00 UTC+9	▶ [System] [Information] [7036] [Service Control Manager] [WIN-BHGRATORDMN] [WinHTTP Web Proxy Auto-Discovery Service サービスは 停止 状態に移行しまし…]	
2015-04-27 07:25:00 UTC+9	▶ [System] [Information] [7036] [Service Control Manager] [WIN-BHGRATORDMN] [WinHTTP Web Proxy Auto-Discovery Service サービスは 停止 状態に移行しまし…]	
2015-04-27 08:02:45 UTC+9	▶ [System] [Information] [7036] [Service Control Manager] [WIN-BHGRATORDMN] [WinHTTP Web Proxy Auto-Discovery Service サービスは 実行中 状態に移行しま…]	
2015-04-27 08:23:38 UTC+9	▶ [System] [Information] [7036] [Service Control Manager] [WIN-BHGRATORDMN] [Windows Modules Installer サービスは 実行中 状態に移行しました。]	
2015-04-27 08:25:39 UTC+9	▶ [System] [Information] [7036] [Service Control Manager] [WIN-BHGRATORDMN] [Windows Modules Installer サービスは 停止 状態に移行しました。]	
2015-04-27 08:31:17 UTC+9	▶ [System] [Information] [7036] [Service Control Manager] [WIN-BHGRATORDMN] [WinHTTP Web Proxy Auto-Discovery Service サービスは 停止 状態に移行しまし…]	
2015-04-27 09:05:23 UTC+9	▶ [System] [Information] [7036] [Service Control Manager] [WIN-BHGRATORDMN] [WinHTTP Web Proxy Auto-Discovery Service サービスは 実行中 状態に移行しま…]	

# CloudWatch Logsのクライアント

- Linuxの場合: テキストログ
  - CloudWatch Logs Agent
- Windowsの場合: テキストログ, Windowsイベントログ、パフォーマンスカウンタ
  - EC2Config
- その他
  - AWS CLI
  - AWS SDK
  - サードパーティツール (fluentd など)

# Log Agentの通信要件

- CloudWatch Logsエンドポイントに接続できること



# Linux:CloudWatch Logs Agent

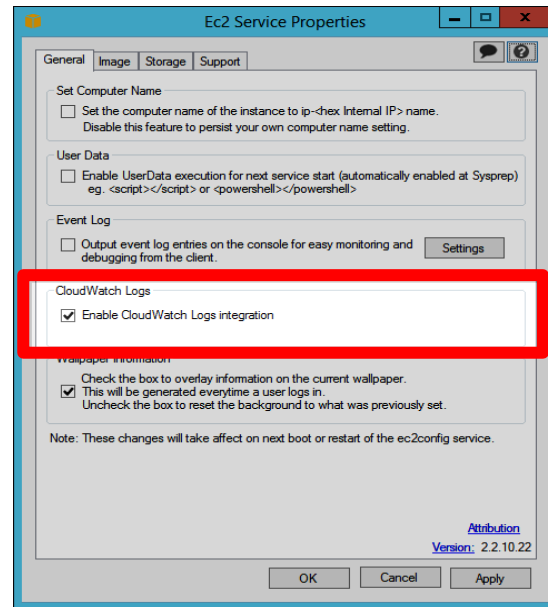
- CloudWatch Logs Agent
  - ログデータをCloudWatch LogsにプッシュするAWS CLIプラグイン
  - CloudWatch Logsにデータを送信するaws logs pushコマンドを実行するスクリプト（デーモン）
  - デーモンが常に実行中であることを確認するcronジョブ
  - インストールと構成が必要
  - 導入対象EC2インスタンスにはRoleをつける
- 対応プラットフォーム
  - Amazon Linux
  - Ubuntu Server
  - CentOS
  - Red Hat Enterprise Linux

# CloudWatch Logs Agentの注意点

- 転送できるログメッセージの長さの制限
  - Agentが1回あたりプッシュできるログレコードサイズは最大32KB
  - 32KBを超えると、ログがトランケートされる
- 対応しているログローテーション
  - rename and re-create
    - 元のログにsuffix(数値)をつけてrenameし、空のログファイルを再作成
    - 例) /var/log/syslog.log が /var/log/syslog.log.1 に 変更される場合
  - copy and truncate
    - 元のログファイルをコピーしてからTRUNCATE
    - 例) /var/log/syslog.log が /var/log/syslog.log.1 に copy され、/var/log/syslog.log が TRUNCATE される場合
  - create common-patterned file
    - 共通のパターンを持つ新しいファイルを作成
    - 例) /var/log/syslog.log.2014-01-01 を残し、 /var/log/syslog.log.2014-01-02 が作成される 場合
    - ※access\_log\_80, access\_log\_443のような異なるファイルに交互に書き込まれるようなファイル監視は未対応

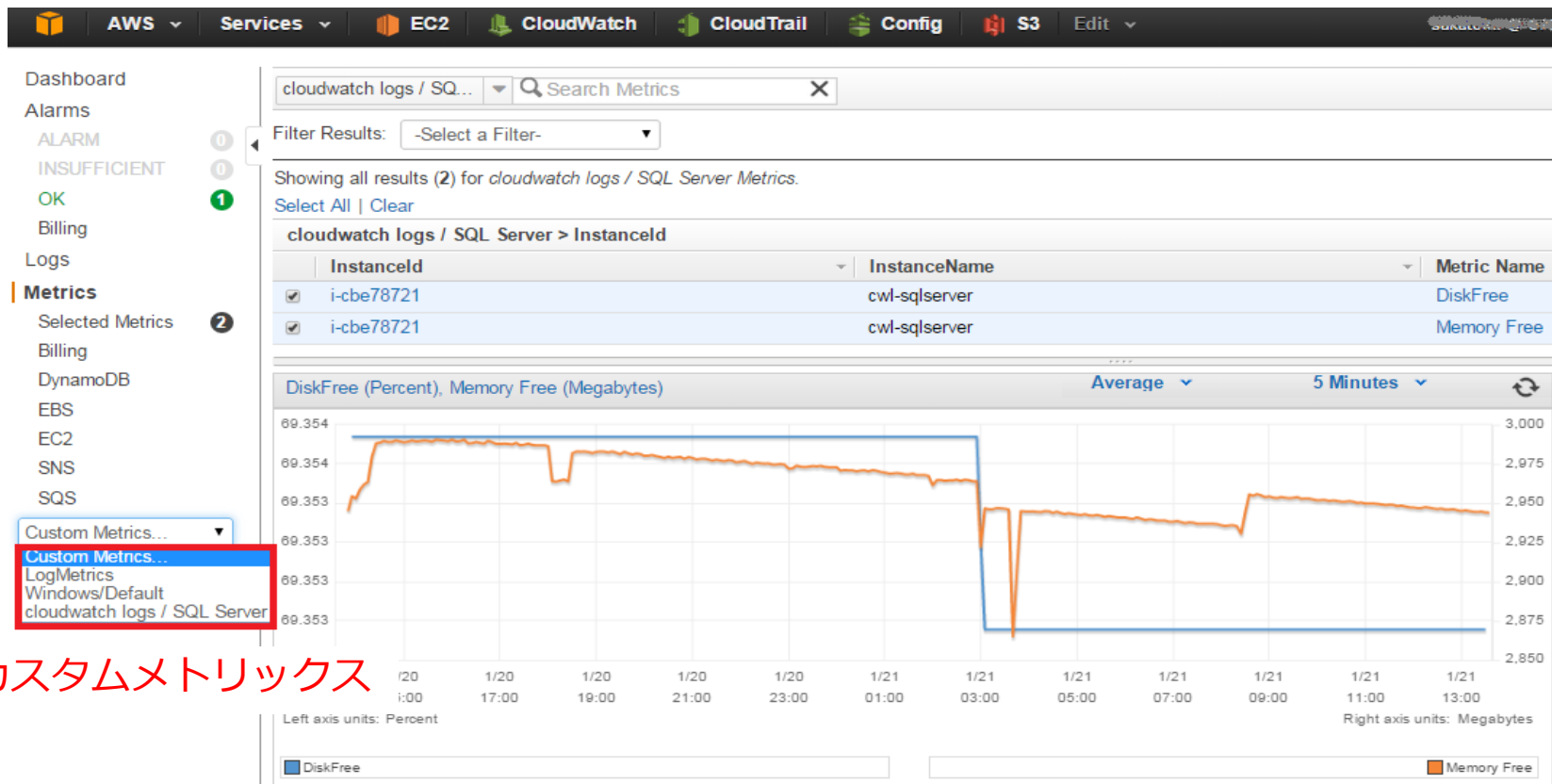
# Windows:CloudWatch Logs設定

- ec2configでCloudWatch Logs設定
  - JSON設定ファイルを編集することで、監視するメトリックスを設定
    - “C:¥Program Files¥Amazon¥Ec2ConfigService¥Settings”に配置されている  
“AWS.EC2.Windows.CloudWatch.json” ファイルを編集
- 収集するログを設定
  - Windowsイベントログ
  - Event Tracing for Windows
  - テキストベース ログ
  - IISログ
  - パフォーマンスカウンタ



チェックを入れるだけで利用可能

# カスタムメトリックスの確認



カスタムメトリックス

# CloudWatch Logs Metric Filter (1/3)

- 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

## Define Logs Metric Filter

### Filter for Log Group: Linux-Sysytem-Logs

You can use metric filters to monitor events in a log group as they are sent to CloudWatch Logs. You can monitor and count specific terms or extract values from log events and associate the results with a metric. [Learn more about pattern syntax.](#)

#### Filter Pattern

DHCPREQUEST

[Show examples](#)

#### Select Log Data to Test

i-156818e7

Test Pattern

[Clear](#)

```
Oct 18 03:01:04 ip-172-31-29-54 dhclient[1878]: DHCPREQUEST on eth0 to 172.31.16.1 port 6
Oct 18 03:01:04 ip-172-31-29-54 dhclient[1878]: DHCPACK from 172.31.16.1 (xid=0x15513168)
Oct 18 03:01:06 ip-172-31-29-54 dhclient[1878]: bound to 172.31.29.54 -- renewal in 1690
Oct 18 03:29:16 ip-172-31-29-54 dhclient[1878]: DHCPREQUEST on eth0 to 172.31.16.1 port 6
Oct 18 03:29:16 ip-172-31-29-54 dhclient[1878]: DHCPACK from 172.31.16.1 (xid=0x15513168)
Oct 18 03:29:18 ip-172-31-29-54 dhclient[1878]: bound to 172.31.29.54 -- renewal in 1585
```

#### Results

Found 17 matches out of 50 event(s) in the sample log.

[Show test results](#)

Cancel

Assign Metric

### Results

Found 17 matches out of 50 event(s) in the sample log.

Line Number	Line Content
1	Oct 18 03:01:04 ip-172-31-29-54 dhclient[1878]: DHCPREQUEST on eth0 to 172.31.16.1 port 67 (xid=0x1551
4	Oct 18 03:29:16 ip-172-31-29-54 dhclient[1878]: DHCPREQUEST on eth0 to 172.31.16.1 port 67 (xid=0x1551
7	Oct 18 03:55:43 ip-172-31-29-54 dhclient[1878]: DHCPREQUEST on eth0 to 172.31.16.1 port 67 (xid=0x1551
10	Oct 18 04:20:18 ip-172-31-29-54 dhclient[1878]: DHCPREQUEST on eth0 to 172.31.16.1 port 67 (xid=0x1551
13	Oct 18 04:43:37 ip-172-31-29-54 dhclient[1878]: DHCPREQUEST on eth0 to 172.31.16.1 port 67 (xid=0x1551
16	Oct 18 05:11:50 ip-172-31-29-54 dhclient[1878]: DHCPREQUEST on eth0 to 172.31.16.1 port 67 (xid=0x1551
19	Oct 18 05:39:11 ip-172-31-29-54 dhclient[1878]: DHCPREQUEST on eth0 to 172.31.16.1 port 67 (xid=0x1551
22	Oct 18 06:06:23 ip-172-31-29-54 dhclient[1878]: DHCPREQUEST on eth0 to 172.31.16.1 port 67 (xid=0x1551
25	Oct 18 06:32:33 ip-172-31-29-54 dhclient[1878]: DHCPREQUEST on eth0 to 172.31.16.1 port 67 (xid=0x1551
29	Oct 18 03:01:04 ip-172-31-29-54 dhclient[1878]: DHCPREQUEST on eth0 to 172.31.16.1 port 67 (xid=0x1551
32	Oct 18 03:29:16 ip-172-31-29-54 dhclient[1878]: DHCPREQUEST on eth0 to 172.31.16.1 port 67 (xid=0x1551
35	Oct 18 03:55:43 ip-172-31-29-54 dhclient[1878]: DHCPREQUEST on eth0 to 172.31.16.1 port 67 (xid=0x1551
38	Oct 18 04:20:18 ip-172-31-29-54 dhclient[1878]: DHCPREQUEST on eth0 to 172.31.16.1 port 67 (xid=0x1551
41	Oct 18 04:43:37 ip-172-31-29-54 dhclient[1878]: DHCPREQUEST on eth0 to 172.31.16.1 port 67 (xid=0x1551
44	Oct 18 05:11:50 ip-172-31-29-54 dhclient[1878]: DHCPREQUEST on eth0 to 172.31.16.1 port 67 (xid=0x1551
47	Oct 18 05:39:11 ip-172-31-29-54 dhclient[1878]: DHCPREQUEST on eth0 to 172.31.16.1 port 67 (xid=0x1551
50	Oct 18 06:06:23 ip-172-31-29-54 dhclient[1878]: DHCPREQUEST on eth0 to 172.31.16.1 port 67 (xid=0x1551

# CloudWatch Logs Metric Filter (2/3)

- 特定文字列のエントリ頻度によりアラーム作成が可能

→ コンソールへのログインが失敗するとアラーム警告

## メトリックスフィルタの作成とメトリックスの割り当て

ロググループ "CloudTrail-Virginia" のフィルタ "cloudwatchlogsalarm-ConsoleSignInFailuresMetricFilter-1VSPC94XPUIISK" の編集

ログインイベントが定義したパターンと一致すると、指定したメトリックスに記録されます。メトリックスをグラフ表示でき、メトリックスにアラームを設定して通知することもできます。

フィルタの名前: cloudwatchlogsalarm-ConsoleSignInFailuresMetricFilter-1VSPC94XPUIISK ⓘ

フィルタパターン: { (\$.eventName = ConsoleLogin) && (\$.errorMessage = "Failed authentication") }

## メトリックスの詳細

メトリックス名前空間: CloudTrailMetrics ⓘ 新しい名前空間の作成

メトリックス名: ConsoleSignInFailureCount ⓘ

メトリックス値: 1 ⓘ

キャンセル

戻る

フィルタの保存

# CloudWatch Logs Metric Filter (3/3)

- Metric Filterからアラーム作成、SNS連携が可能

Log Groups > Filters for Linux-Sysytem-Logs

Add Metric Filter

✓ Your filter **error message filtering** has been created.

Filter Name: error message filtering  
Filter Pattern: error  
Metric: LogMetrics / error  
Metric Value: 3

Create Alarm

Create Alarm

1. Select Metric 2. Define Alarm

Alarm Threshold

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

Name: CWL Linux SystemLog Errors

Description: CWL Linux SystemLog Errors

Whenever: error

is:  $\geq$  0

for: 1 consecutive period(s)

Actions

Define what actions are taken when your alarm changes state.

Notification

Delete

Whenever this alarm: State is ALARM

Send notification to: NotifyMe

New list Enter list

Email list:

sakatoku@amazon.co.jp

Email

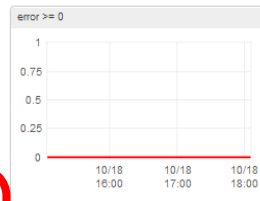
+ Notification

+ AutoScaling Action

+ EC2 Action

Alarm Preview

This alarm will trigger when the blue line goes up to or above the red line for a duration of 1 minute



Namespace: LogMetrics

Metric Name: error

Period: 1 Minute

Statistic: Sum

Metric FilterをトリガーにしたCloudWatch  
アラームの作成が可能

# メトリックスフィルタ (一般的なログフォーマット)

```
127.0.0.1 - frank [10/Oct/2000:13:25:15 -0700] "GET /apache_pb.gif HTTP/1.0" 200 1534
127.0.0.1 - frank [10/Oct/2000:13:35:22 -0700] "GET /apache_pb.gif HTTP/1.0" 500 5324
127.0.0.1 - frank [10/Oct/2000:13:50:35 -0700] "GET /apache_pb.gif HTTP/1.0" 200 4355
```

- Filter example

[ip, user, username, timestamp, request, status\_code, bytes > 1000]

[ip, user, username, timestamp, request, status\_code = 200, bytes]

[ip, user, username, timestamp, request, status\_code = 4\*, bytes]

[ip, user, username, timestamp, request = \*html\*, status\_code = 4\*, bytes]

- Notes

- 角括弧 [] または 2 個の二重引用符 ("" ) で囲まれた文字は単一のフィールドとして扱われます
- メトリックスフィルタは大文字と小文字を区別します

# CloudWatch Logs料金体系

	無料枠 (1カ月あたり)	従量課金
インジェクション	5GB	\$0.76 / GB
アーカイブ	5GB	\$0.033 GB / 月

10メトリックス、10アラーム、および100万の API リクエストの無料利用枠を準備

<http://aws.amazon.com/jp/cloudwatch/pricing/>

# CloudWatch Logs検索機能の向上

- 複数のログストリームにまたがってログを閲覧したり、キーワード検索が可能（1つのロググループ内）

ログストリームを複数選択し、  
イベントを検索ボタンをクリック

ロググループ > /var/log/httpd/access\_log のストリーム

イベントを検索 ログストリームの作成 ログストリームの削除

フィルタ: ログストリーム名のプレフィックス: x ログストリーム 1-2

ログストリーム 直前のイベント時刻

ログストリーム	直前のイベント時刻
i-xxxxxxx	2015-06-24 17:10 UTC+9
i-xxxxxxx	

ロググループ > /var/log/httpd/access\_log のストリーム

ログストリームをフィルタ 検索キーワード

フィルタ: Wget

日付/時刻: 2015/06/24 16:47:29 ローカル (GMT+09:00)

ログストリーム名	イベントデータ
i-xxxxxxx	127.0.0.1 -- [24/Jun/2015:07:45:43 +0000] "GET / HTTP/1.1" 403 3839 "-" "Wget/1.16.1 (linux-gnu)"
i-xxxxxxx	127.0.0.1 -- [24/Jun/2015:07:47:58 +0000] "GET / HTTP/1.1" 403 3839 "-" "Wget/1.16.1 (linux-gnu)"
i-xxxxxxx	127.0.0.1 -- [24/Jun/2015:08:03:25 +0000] "GET / HTTP/1.1" 403 3839 "-" "Wget/1.16.1 (linux-gnu)"
i-xxxxxxx	127.0.0.1 -- [24/Jun/2015:08:05:18 +0000] "GET /index.html HTTP/1.1" 200 36 "-" "Wget/1.16.1 (linux-gnu)"
i-xxxxxxx	192.168.100.153 -- [24/Jun/2015:08:05:34 +0000] "GET / HTTP/1.1" 200 36 "-" "Wget/1.16.1 (linux-gnu)"
i-xxxxxxx	192.168.101.212 -- [24/Jun/2015:08:10:11 +0000] "GET / HTTP/1.1" 200 36 "-" "Wget/1.16.1 (linux-gnu)"

# CloudWatch Logs Subscription

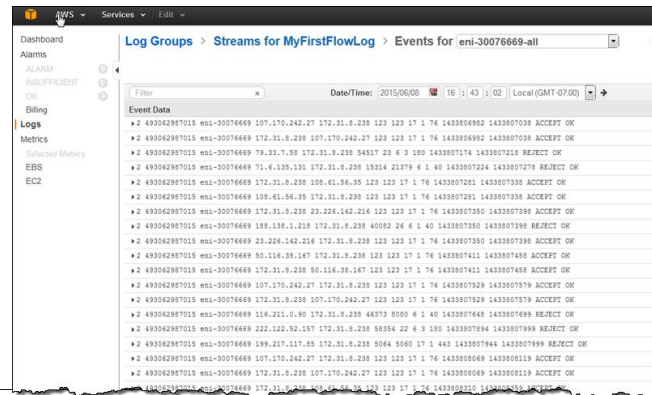
- CloudWatch Logsに集めたログをリアルタイムにKinesisに転送可能に
  - Kinesis Applicationでログに対して／を受けてロジックを記述可能
  - CloudWatch Logsには指定した保管期間ログが保管される



```
aws logs put-subscription-filter ¥  
  --log-group-name "xxxxxxx" ¥  
  --filter-name "xxxxxxx" ¥  
  --filter-pattern "{xxxxxxx = xxxxxxx}" ¥  
  --destination-arn "arn:aws:kinesis:ap-northeast-1:123456789012:stream/xxxxxxx" ¥  
  --role-arn "arn:aws:iam::123456789012:role/xxxxxxx"
```

# VPC Flow Logs

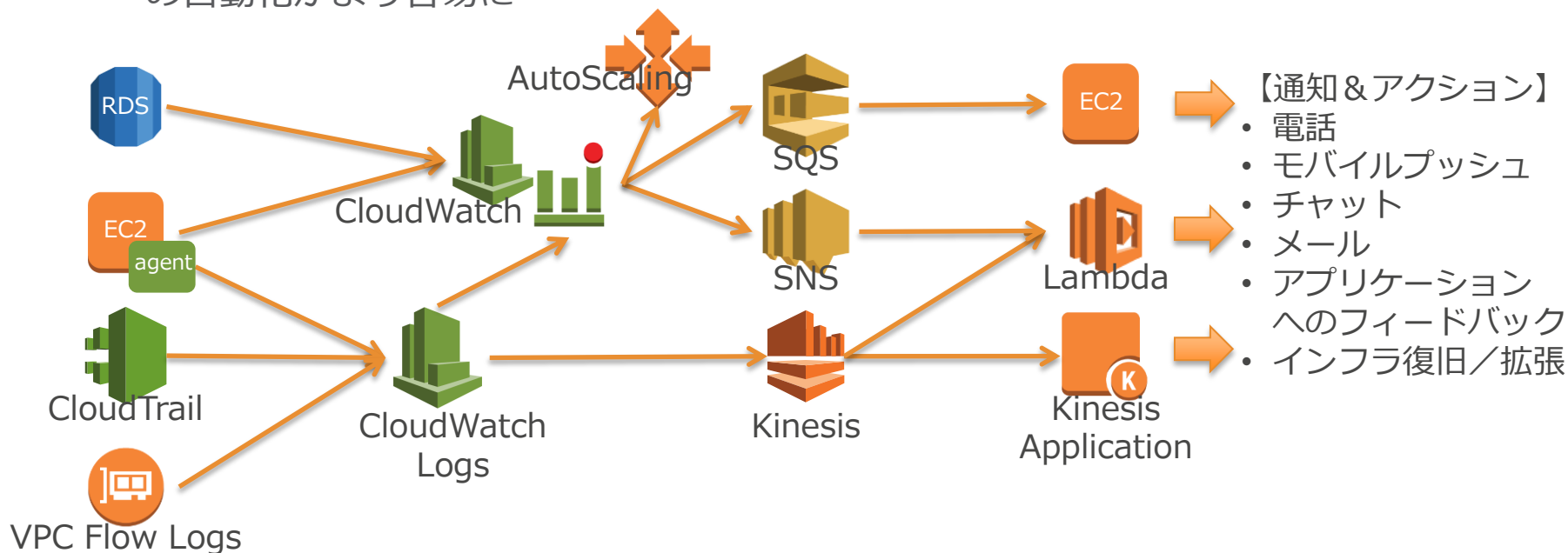
- VPC内ネットワーク・ログ
  - VPC内のネットワークのログが取得可能
  - Network ACLとSecurity Groupでの許可と禁止のトラフィックログ
  - VPC Flow Logsの設定は、VPC、Subnet、ENIに対して実施
  - ログは、ネットワークインタフェース（ENI）ごとに取得（ENIがログストリーム）
  - 特定の通信はログされない
    - Amazon Provided DNSへのDNS Lookup、Windowsライセンスアクティベーション、DHCP等



# CloudWatchを介したシステム運用の自動化

- システム運用の自動化

- CloudWatchおよびCloudWatch Logsの各種連携機能が登場し、システム運用の自動化がより容易に



# CloudWatchを使った自動化例

- IAMの設定変更に関するアラートの受け取り方について
  - [http://aws.typepad.com/aws\\_japan/2015/02/how-to-receive-alerts-when-your-iam-configuration-changes.html](http://aws.typepad.com/aws_japan/2015/02/how-to-receive-alerts-when-your-iam-configuration-changes.html)
- AWSアカウントのルートアクセスキーを使用した場合の通知方法
  - [http://aws.typepad.com/aws\\_japan/2015/06/how-to-receive-notifications-when-your-aws-accounts-root-access-keys-are-used.html](http://aws.typepad.com/aws_japan/2015/06/how-to-receive-notifications-when-your-aws-accounts-root-access-keys-are-used.html)
- CloudWatch Logs Subscriptionsを利用したZabbixへのログ転送
- アラート発生時に AWS Lambda を使って音声電話をかける
- CloudWatchのAlertをAWS Lambda経由でSlackに飛ばす
- Auto Scalingによる自動復旧（AWS Lambda+SNS編）
- Amazon ECSのDockerコンテナをLambdaでAuto Scalingに連携させる
- VPC Flow LogsをElasticsearch + Kibana4で可視化する



# AWSでの運用監視

# AWSの運用監視

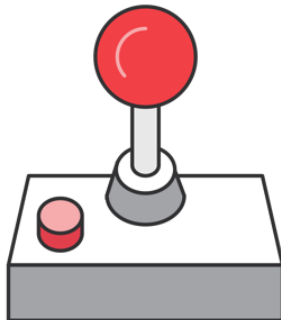
- 今までのシステム監視と変わらない
  - オンプレミス時の運用ノウハウを最大限活用
  - AWSサービスをうまく活用したシンプルな監視
  - 多くの監視ツールがAWSに対応
- クラウドならではの監視
  - スケール監視
  - コスト監視
  - コンプライアンス



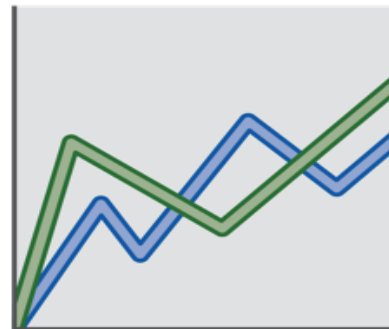
# 基本となる運用監視



性能監視



死活監視



キャパシティ監視

オンプレミスでもクラウドでも基本的には同じ

# 監視ツール連携の必要性

- サードパーティ監視ツールの必要性
  - ハイブリッド環境での統合監視
  - CloudWatch機能制限への対応
    - プロセス監視
    - ログ監視での柔軟な文字列マッチング
    - メンテナンスウィンドウの設定
    - 重要度の設定
    - 長期保管
- 環境と監視項目ごとのツール利用例

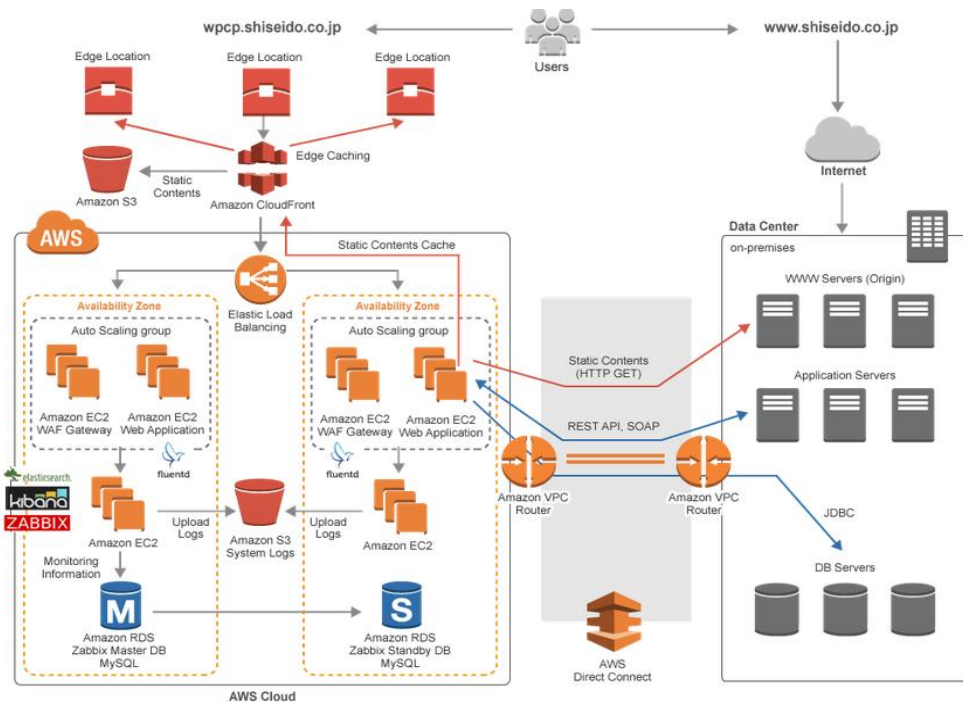


	AWS		オンプレミス
	EC2	ELB/RDS	サーバ
性能監視 (CPU使用率、DiskI/O他)	CloudWatch標準メトリックス or サードパーティツール	CloudWatch標準メトリックス	サードパーティツール
死活監視 (ホスト)	CloudWatch標準メトリックス or サードパーティツール	CloudWatch標準メトリックス	サードパーティツール
死活監視 (プロセス)	サードパーティツール	CloudWatch標準メトリックス	サードパーティツール
キャパシティ監視 (メモリ使用、ディスク使用量)	CloudWatchカスタムメトリックス or WindowsであればCloudWatch Logs or サードパーティツール	CloudWatch標準メトリックス	サードパーティツール

# 監視システムとのAmazon CloudWatch連携



## 監視システム利用イメージ



## サードパーティ監視ツールの確認ポイント

- AWSに対応しているか
- CloudWatchとの連携機能の有無
- CloudWatchカスタムメトリックスに対応しているか
- Auto Scaling対応しているか
- EC2インスタンス自動検出・自動削除が可能か

ZABBIX

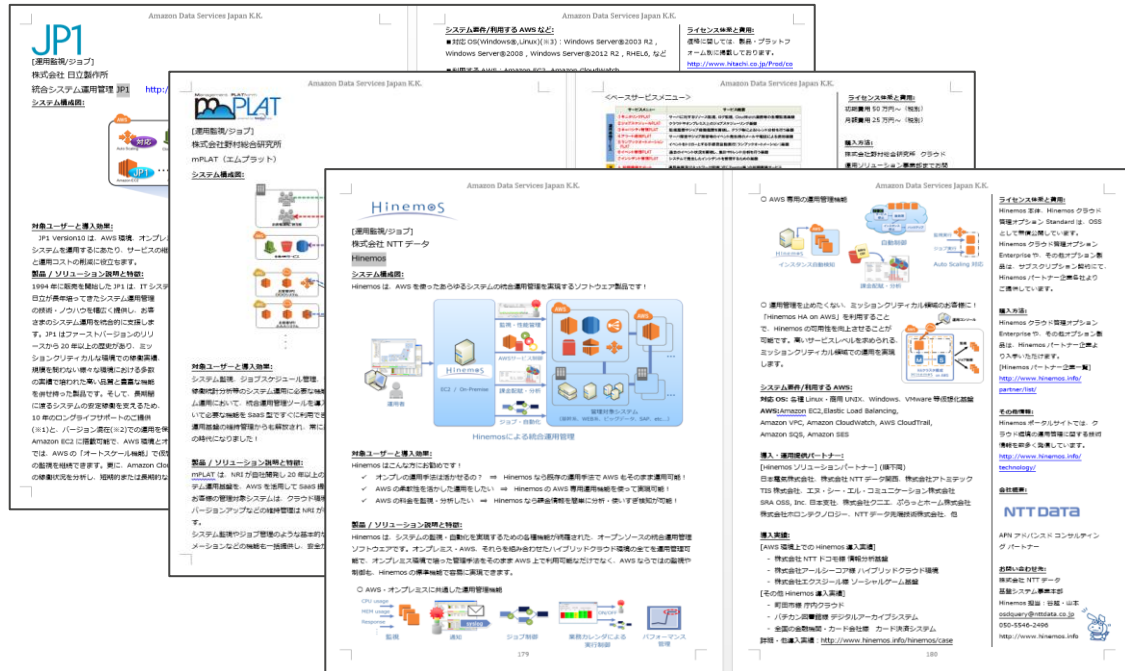
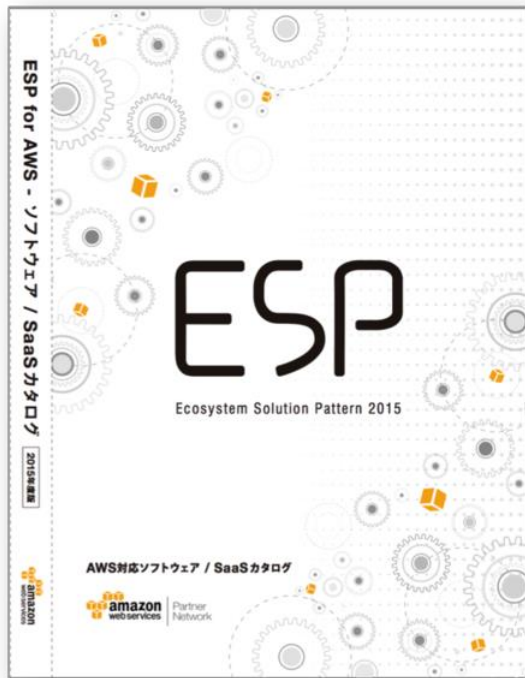
HinemoS

Management PLATFORM  
m PLAT  
powered by Senjyu

JP1

<http://aws.amazon.com/jp/solutions/case-studies/shiseido/>

# ESP(Ecosystem Solution Pattern)カタログ 無料配布 2015年度版 AWS対応ソフトウェア/SaaSガイド

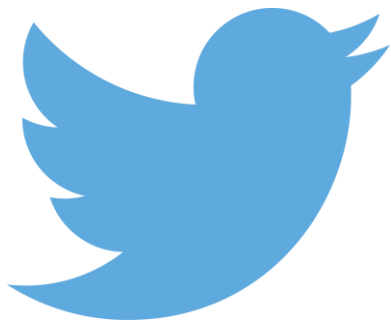


[http://aws.typepad.com/aws\\_japan/2015/06/esp-catalog-pdf.html](http://aws.typepad.com/aws_japan/2015/06/esp-catalog-pdf.html)

# AWS運用コミュニティ

～クラウドによる、クラウドのための、クラウド運用管理～

AWS上に構築されたシステムの  
運用管理のベストプラクティスを集約！



@opsjaws



[http://aws.typepad.com/aws\\_partner\\_sa/2015/06/aws-ops.html](http://aws.typepad.com/aws_partner_sa/2015/06/aws-ops.html)

# Q&A



次回Webinarのお申し込み

[http://aws.amazon.com/jp/event\\_schedule/](http://aws.amazon.com/jp/event_schedule/)

# Webinar資料の配置場所

- AWS クラウドサービス活用資料集
  - <http://aws.amazon.com/jp/aws-jp-introduction/>

プロダクト別：				
Amazon S3		AWSマイスターシリーズ Re:Generate Amazon Simple Storage Service (S3)	Slideshare	PDF
Amazon Glacier		AWSマイスターシリーズ Reloaded Amazon Glacier  Amazon Glacierのご紹介 機能編	Slideshare (Reloaded)  Slideshare (機能編)	PDF (Reloaded)  PDF (機能編)
Amazon Route 53		AWSマイスターシリーズ Re:Generate	Slideshare	PDF

# 公式Twitter/Facebook AWSの最新情報をお届けします



@awscloud\_jp



検索



もしくは  
<http://on.fb.me/1vR8yWm>

最新技術情報、イベント情報、お役立ち情報、お得なキャンペーン情報などを  
日々更新しています！

**ご参加ありがとうございました。**