



AWS Summit

Tokyo



AWS System Operation Deep Dive

Monitoring / Logging / Configuration

TA-09: Tech Deep Dive by Amazon

■ Gold Sponsors



Empowered by Innovation



■ Global Sponsors



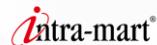
■ Silver Sponsors



野村総合研究所



■ Bronze Sponsors



■ Global Tech Sponsors



■ Logo Sponsors



ハッシュタグ **#AWSSummit**
で、皆さんのツイートが展示エリア
の大画面に表示されます



公式アカウント **@awscloud_jp**
をフォローすると、ロゴ入り
コースターをプレゼント



【コースター配布場所】

メイン展示会場、メイン会場1F受付、デベロッパーカンファレンス会場





AWS System Operation Deep Dive

Monitoring / Logging / Configuration

TA-09: Tech Deep Dive by Amazon

自己紹介

酒徳 知明(さかとく ともあき)

エコシステム ソリューション部

パートナー ソリューション アーキテクト

- エンタープライズ SIパートナー様のご支援
- ISVパートナー様のご支援



好きなAWSサービス

- Amazon CloudWatch, AWS CloudTrail, AWS Config

Agenda

- システム運用で考えるポイント
- システム運用を支えるAWSサービス
- AWSサービスの利用例
- まとめ

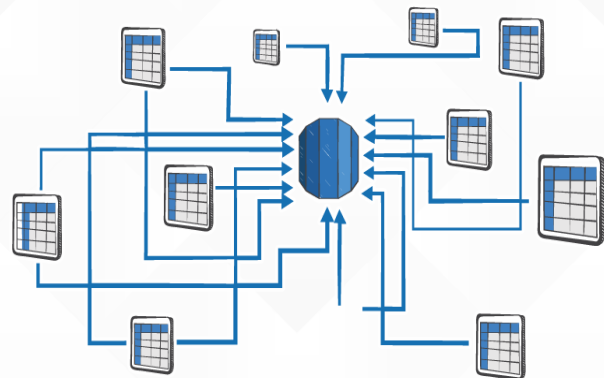
システム基盤運用で考えるポイント



Monitoring
監視



Logging
ロギング



Configuration
構成管理

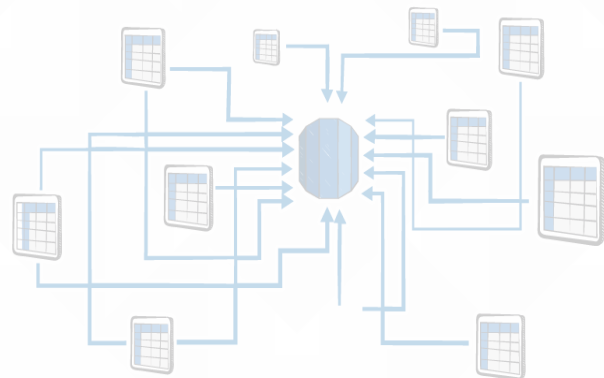
システム基盤運用で考えるポイント



Monitoring
監視



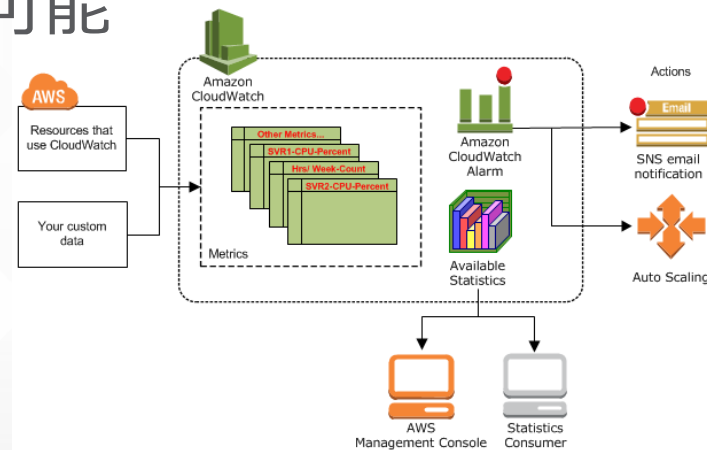
Logging
ロギング



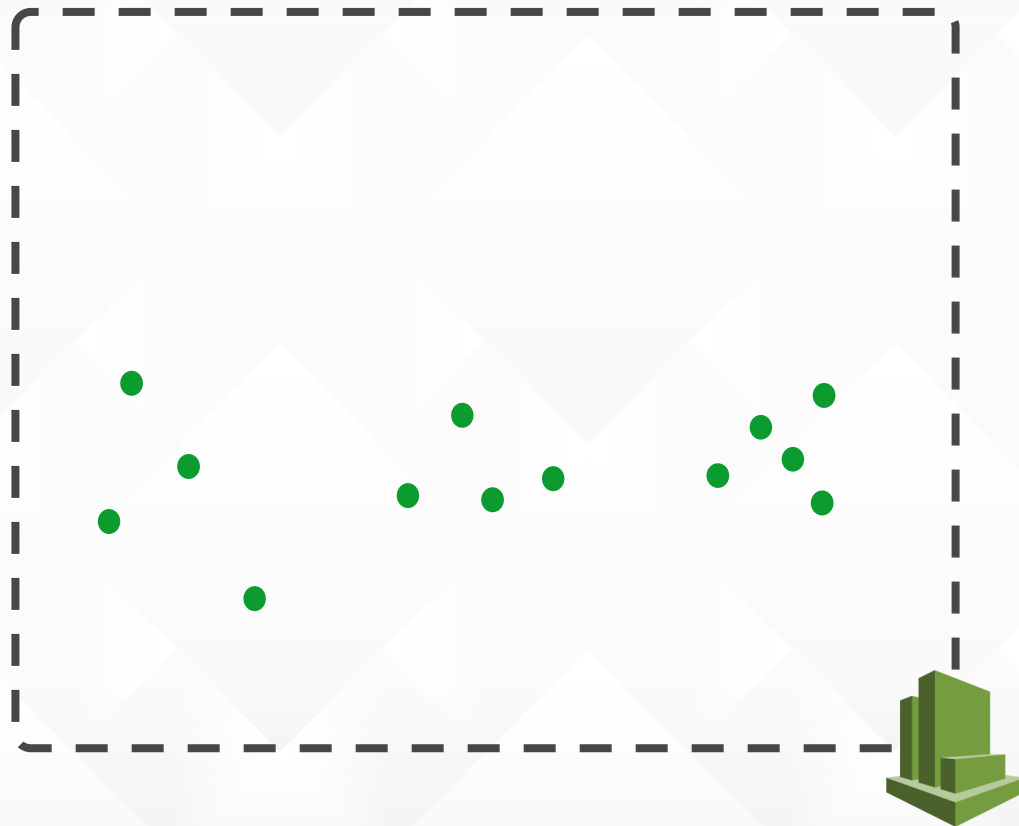
Configuration
構成管理

Amazon CloudWatchの概要

- 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
- 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
- AWSの課金情報の監視



Amazon CloudWatchのコンセプト



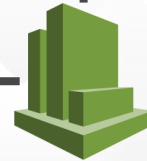
Amazon CloudWatchのコンセプト

Metrics

Metrics Name
(CPUUtilization)



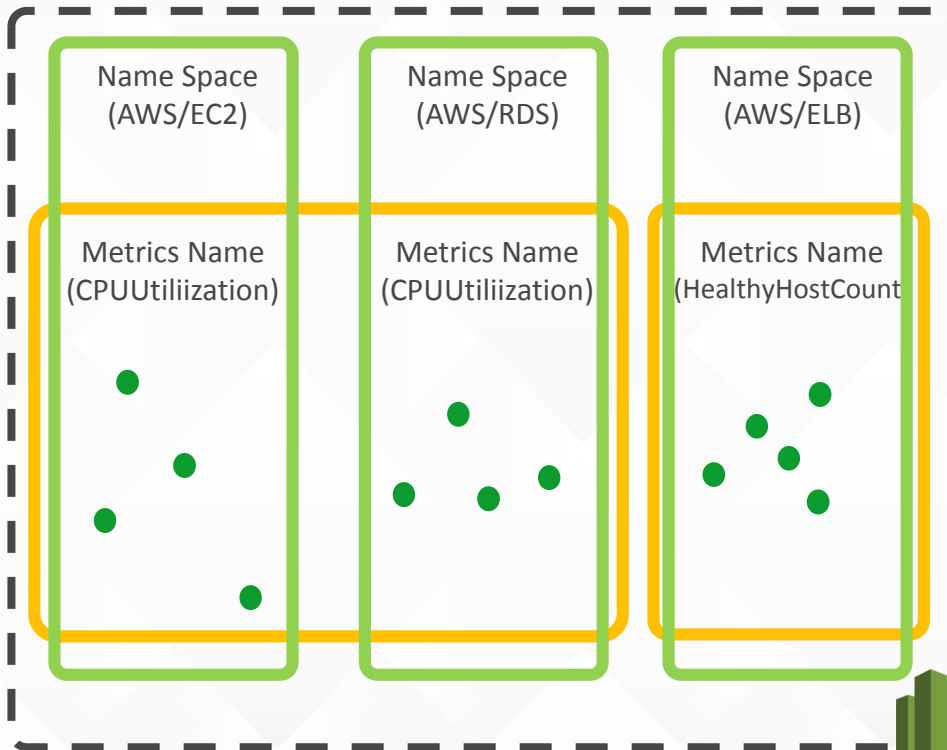
Metrics Name
(HealthyHostCount)



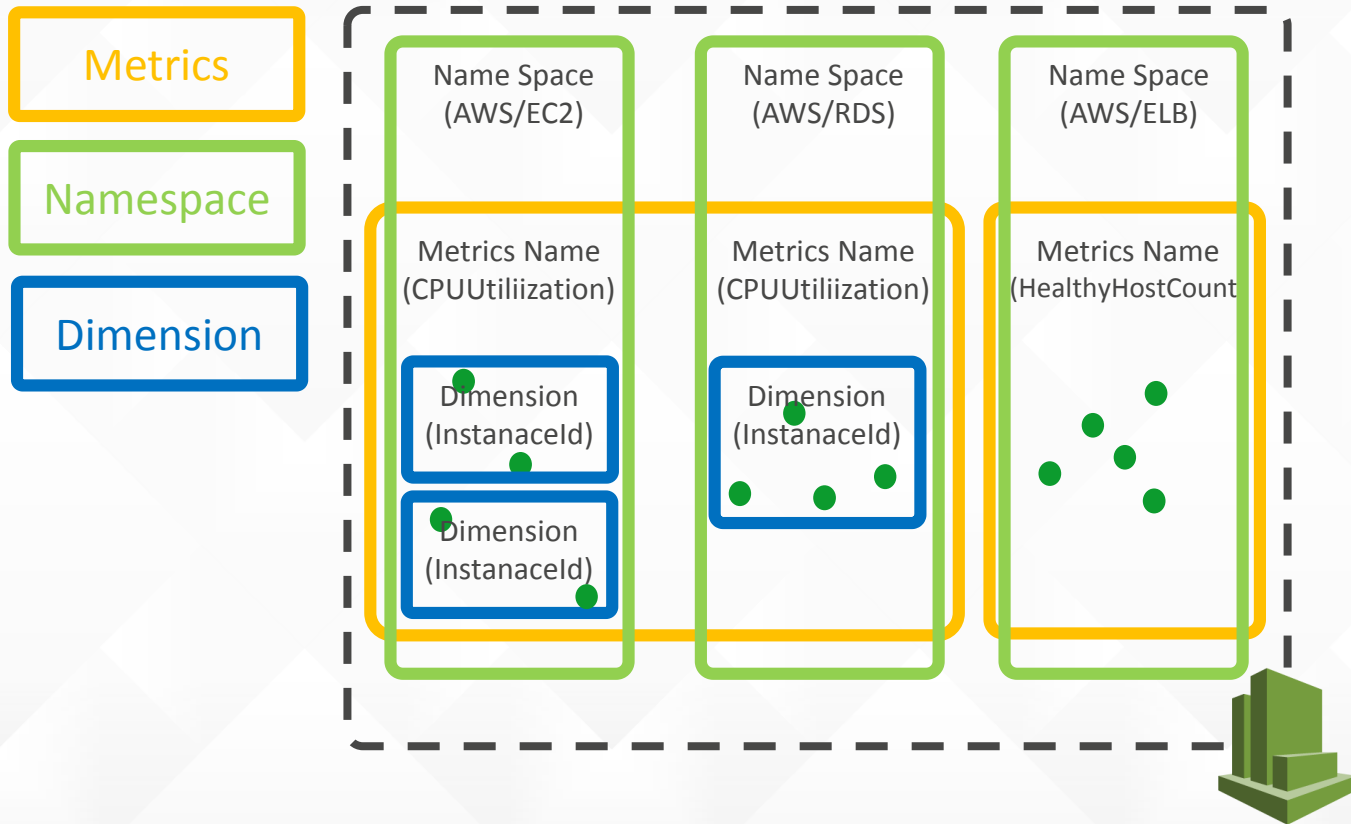
Amazon CloudWatchのコンセプト

Metrics

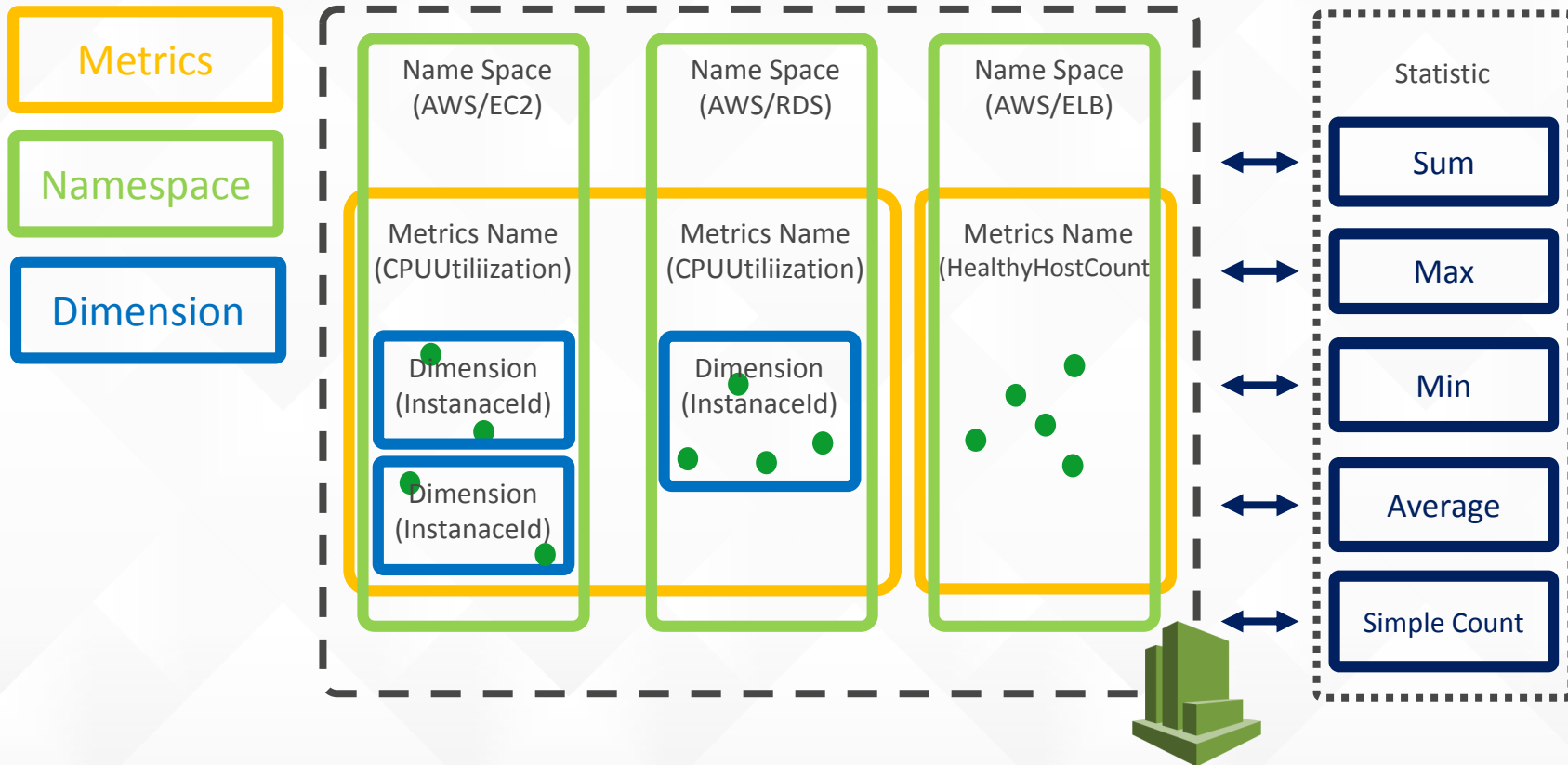
Namespace



Amazon CloudWatchのコンセプト



Amazon CloudWatchのコンセプト



Amazon CloudWatchに対応するAWSサービス

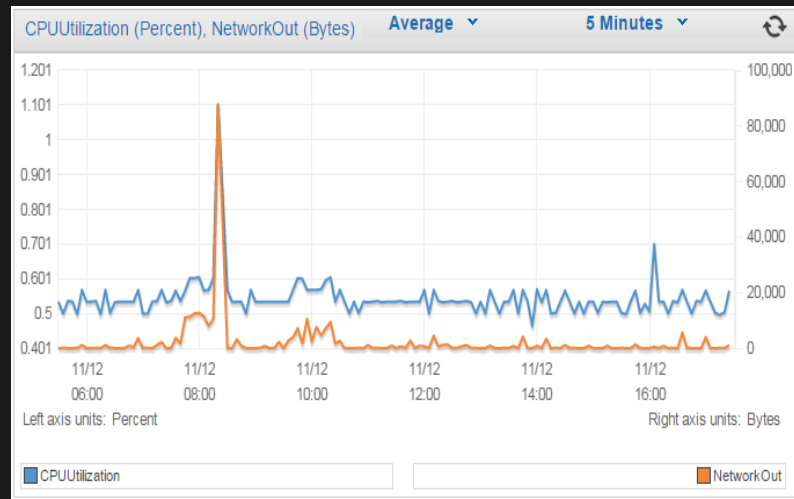
| AWS サービス | 名前空間 |
|------------------------------------|----------------------|
| Auto Scaling | AWS/AutoScaling |
| AWS Billing | AWS/Billing |
| Amazon CloudFront | AWS/CloudFront |
| Amazon CloudSearch | AWS/CloudSearch |
| Amazon DynamoDB | AWS/DynamoDB |
| Amazon ElastiCache | AWS/ElastiCache |
| 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 |
| AWS OpsWorks | AWS/OpsWorks |
| Amazon Redshift | AWS/Redshift |
| Amazon Relational Database Service | AWS/RDS |

| AWS の製品 | 名前空間 |
|------------------------------------|--------------------|
| 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 |

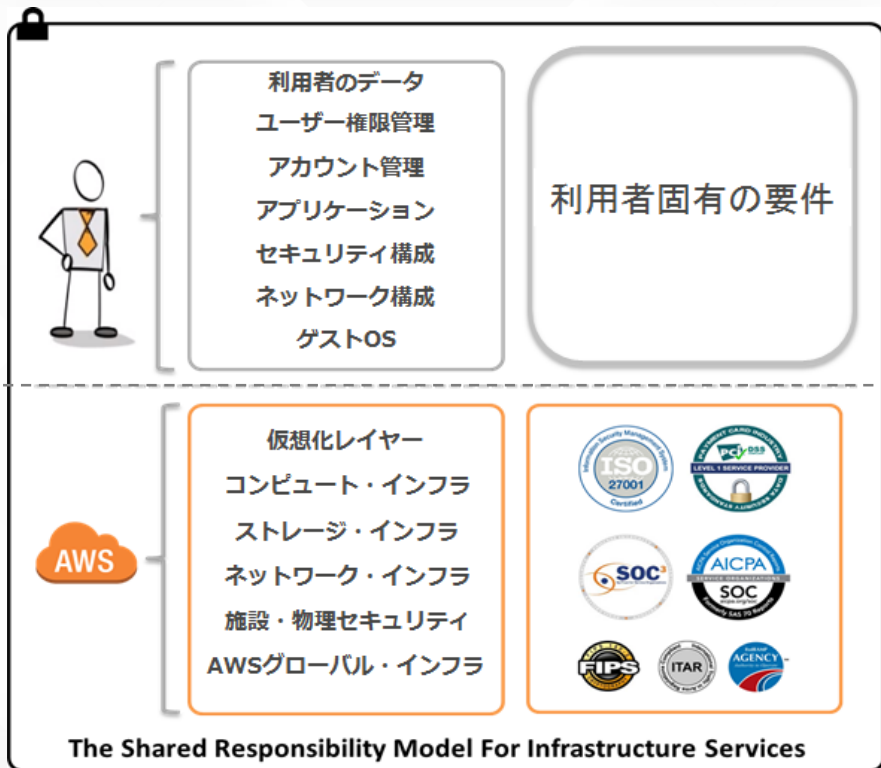
Amazon CloudWatchを使ったモニタリング

```
$ aws cloudwatch get-metric-statistics --metric-name CPUUtilization --start-time 2015-05-19T13:00:00.000Z --end-time 2015-05-19T14:00:00.000Z --period 600 --statistics "Average" --namespace "AWS/EC2" --dimensions Name=InstanceId,Value=i-1e02XXXX
```

```
{  
  "Datapoints": [  
    {  
      "Timestamp": "2015-05-19T13:00:00Z",  
      "Average": 0.033,  
      "Unit": "Percent"  
    },  
    {  
      "Timestamp": "2015-05-19T13:50:00Z",  
      "Average": 0.017,  
      "Unit": "Percent"  
    }  
  ],  
}
```



監視システムとのAmazon CloudWatch連携



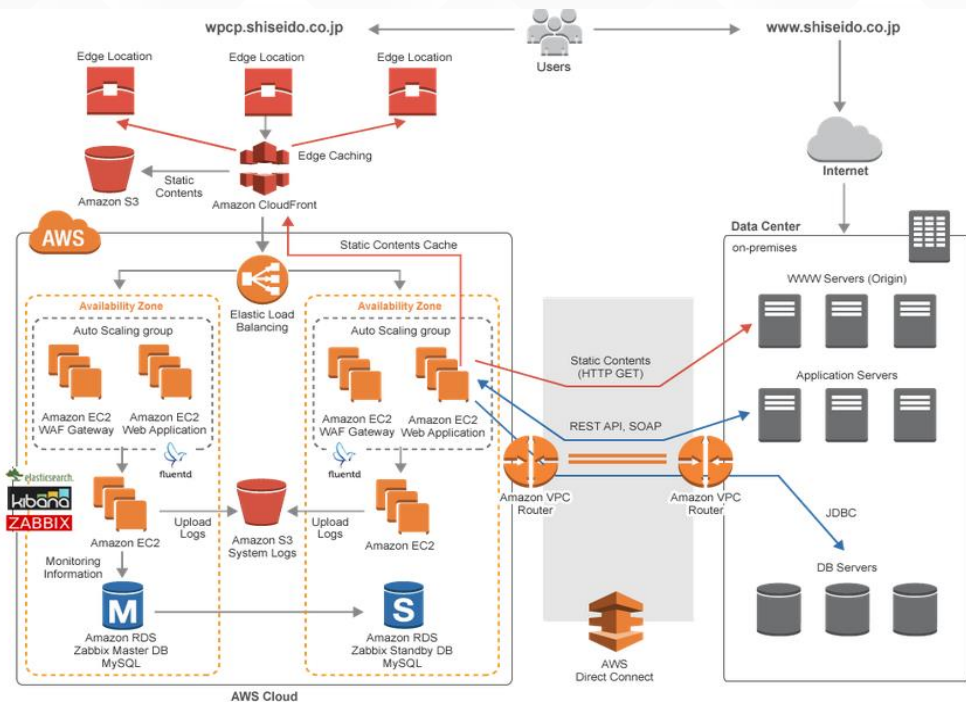
3rd Party 監視ツール



監視システムとのAmazon CloudWatch連携



監視システム利用イメージ



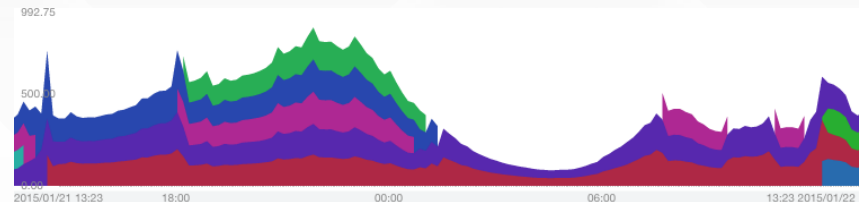
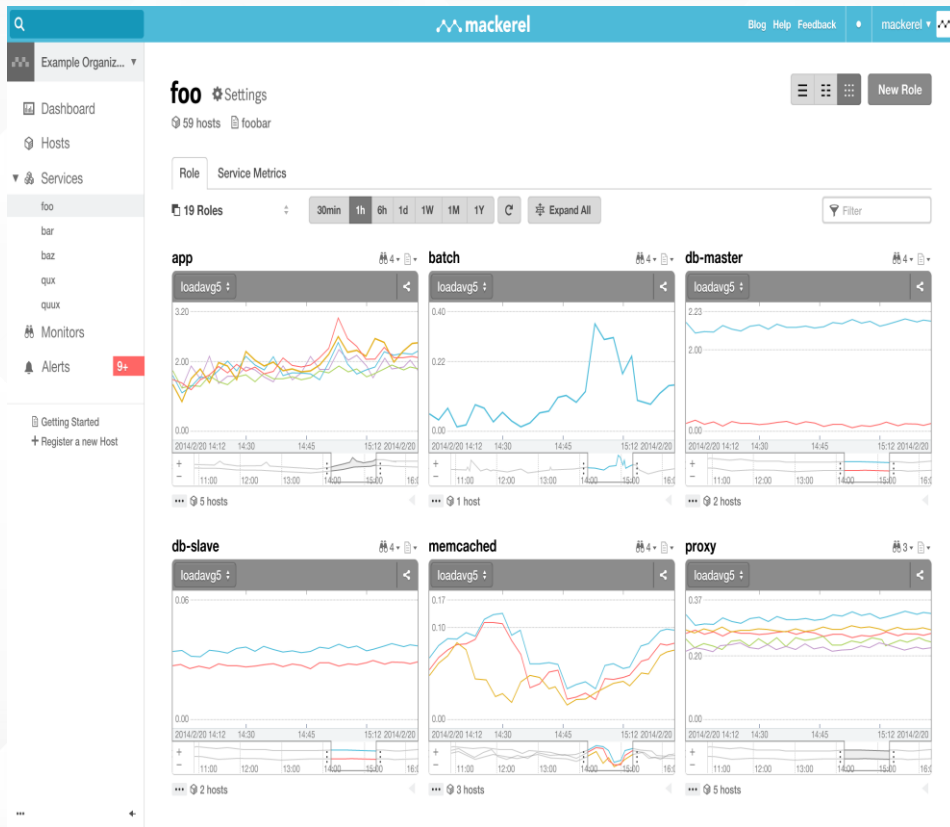
サードパーティ監視ツールの確認ポイント

- 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



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

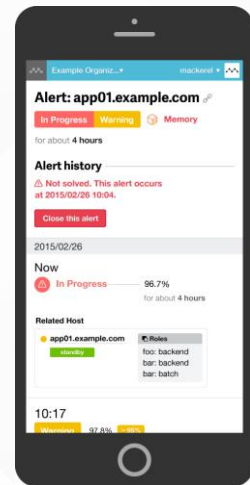
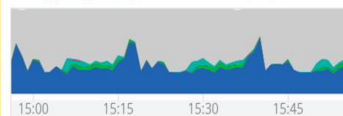
Mackerelを使ったモニタリング



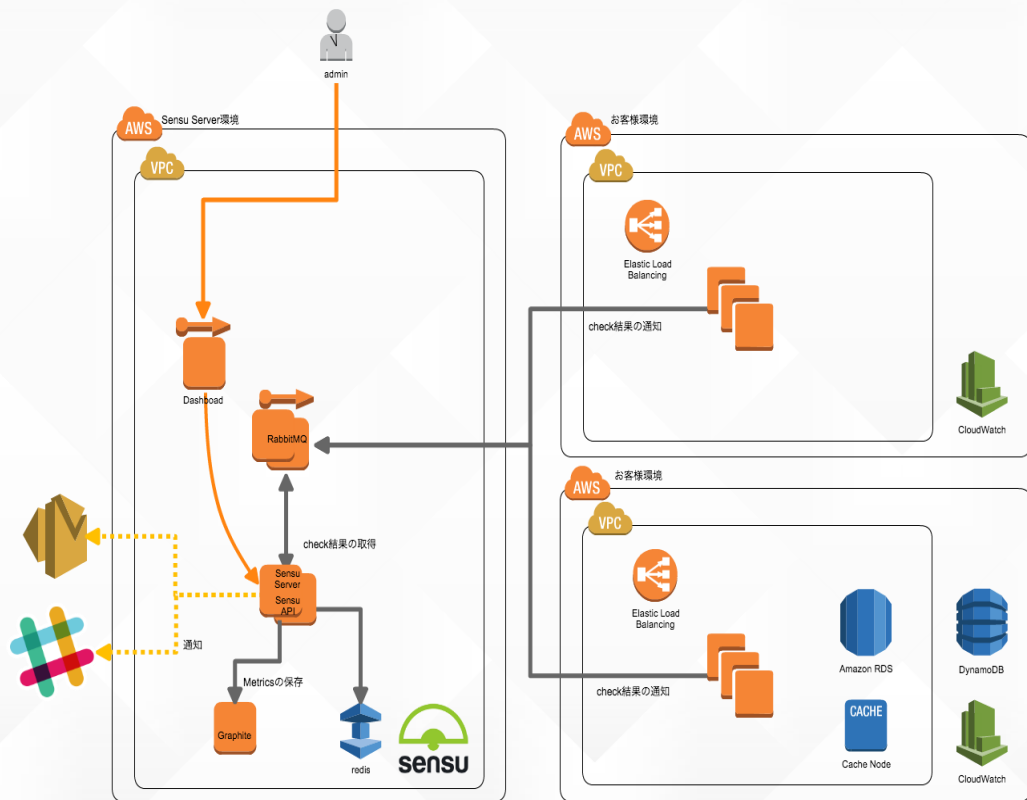
Mackerel BOT 3:55 PM ★
CRITICAL: myblog.example.org loadavg5 5.04 > 5.0 LINK @channel



WARNING: dev [develop.example.org] CPU % develop 41.59% > 40.0% LINK @channel

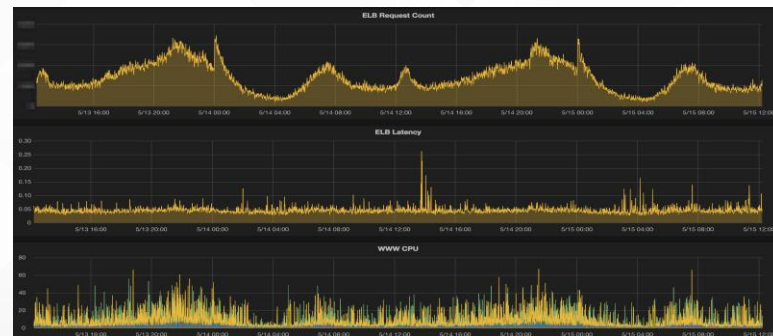


Sensuを使ったモニタリング

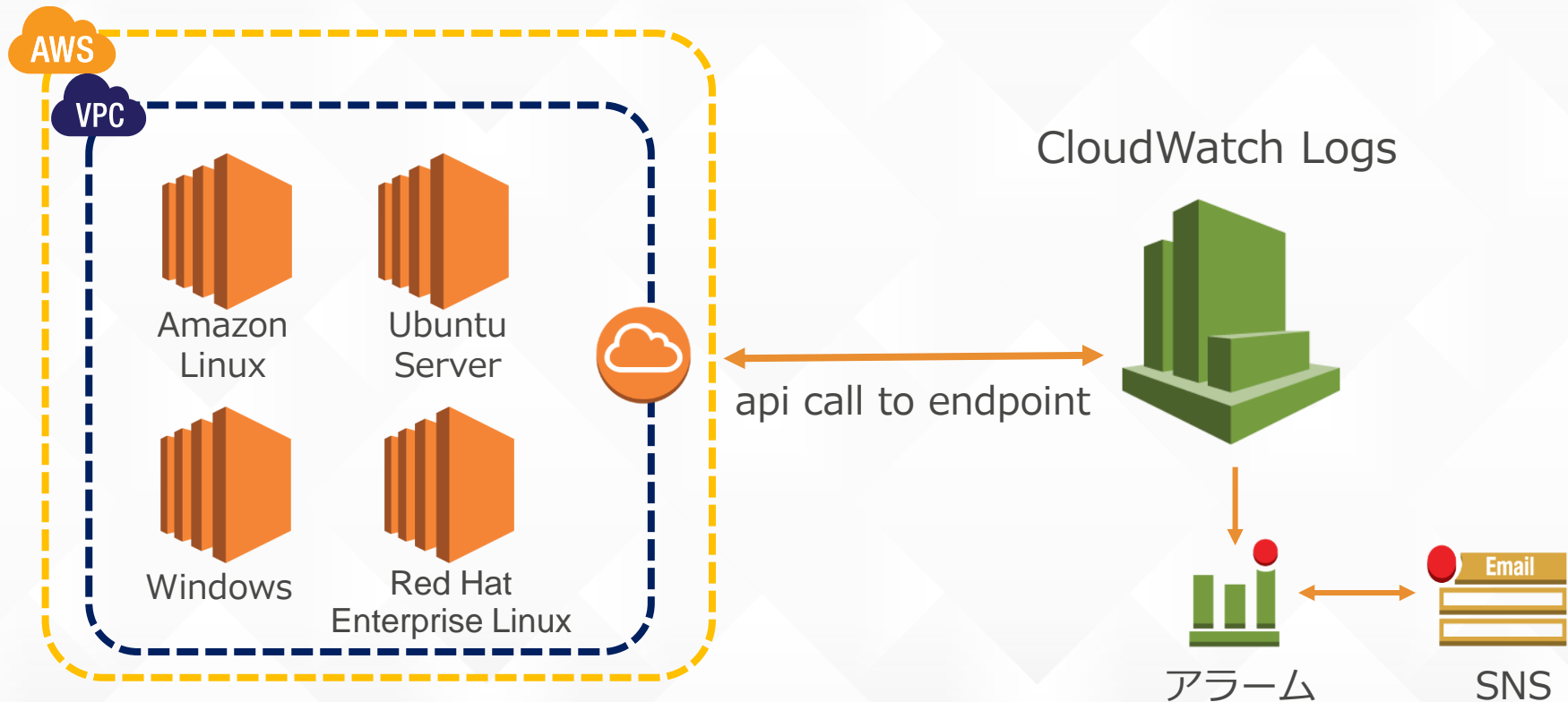


The screenshot shows the **uchiwa** **Clients** page. It displays a list of clients with columns for Name, IP, Events, and other details. The search filter is set to **kappa**.

| Name | IP | Events | Created | Updated | Deleted |
|--------------------------------|----------------|---|------------|---------|---------------------|
| kappa-chef_client01_j-f701fa02 | 52.68.3.40 | No keepalive sent from client for 2588586 seconds (>=180) | gree-sensu | 0.17.0 | 2015-04-15 13:19:21 |
| kappa-chef_client01_j-b3f00f46 | 54.199.129.166 | No keepalive sent from client for 4821813 seconds (>=180) and 1 more... | gree-sensu | 0.16.0 | 2015-03-20 16:58:54 |
| kappa-chef_client01_j-a0f00f46 | 54.199.194.45 | No keepalive sent from client for 4821817 seconds (>=180) and 2 more... | gree-sensu | 0.16.0 | 2015-03-20 16:58:50 |
| kappa-chef_client01_j-a0f00f58 | 54.199.193.248 | No keepalive sent from client for 4821805 seconds (>=180) and 1 more... | gree-sensu | 0.16.0 | 2015-03-20 16:59:02 |
| kappa-chef_client01_j-f937e50c | 54.64.149.60 | No keepalive sent from client for 1876391 seconds (>=180) | gree-sensu | 0.17.0 | 2015-04-23 19:09:16 |
| kappa-chef_client01_j-a2f00f57 | 54.199.194.18 | No keepalive sent from client for 4821799 seconds (>=180) and 1 more... | gree-sensu | 0.16.0 | 2015-03-20 16:59:08 |
| kappa-chef_client01_j-a3f00f56 | 54.199.194.48 | No keepalive sent from client for 4821807 seconds (>=180) and 1 more... | gree-sensu | 0.16.0 | 2015-03-20 16:59:08 |
| kappa-chef_client01_j-da0f002f | 54.199.194.32 | No keepalive sent from client for 4821809 seconds (>=180) and 1 more... | gree-sensu | 0.16.0 | 2015-03-20 16:59:08 |
| kappa-chef_client01_j-b2f00f47 | 54.199.194.39 | No keepalive sent from client for 4821811 seconds (>=180) and 1 more... | gree-sensu | 0.16.0 | 2015-03-20 16:58:56 |
| kappa-chef_client02_j-e637e... | 54.178.188.75 | No keepalive sent from client for 4778286 seconds (>=180) | gree-sensu | 0.16.0 | 2015-03-21 05:04:21 |



CloudWatch Logs



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

/var/log/awslogs/log

```
2014-10-18 10:29:28,770 - cwlogs.push - WARNING - 31955 - Thread-6 -  
Truncate event: {'source_id': '4c0bbd10e46d9c06707aada0cd3e2cd8', 'timestamp':  
None, 'start_position': 748331L, 'end_position': 1748331L},  
reason: single event exceeds 32KB limit.
```


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

※データを損失する恐れがあるため、このファイル更新機能の使用には注意が必要

- 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

ログの転送設定

```
$ sudo cat /var/awslogs/etc/awslogs.conf
[general]
state_file = <value>

[logstream1]
log_group_name = <value>
log_stream_name = <value>
datetime_format = <value>
time_zone = [LOCAL|UTC]
file = <value>
file_fingerprint_lines = <integer> | <integer-integer>
multi_line_start_pattern = <regex> |
{datetime_format}
initial_position = [start_of_file|end_of_file]
encoding = [ascii|utf_8|..]
buffer_duration = <integer>
...
```

file =

- /var/log/system.log* のようにワイルドカードが利用可能
- 圧縮ファイルは未サポート

file fingerprint lines =

- ファイルを識別するために行範囲を指定
- デフォルトでは1行目のみを使いファイルを識別

multi line start pattern =

- ログメッセージの開始を識別するパターンを指定

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]

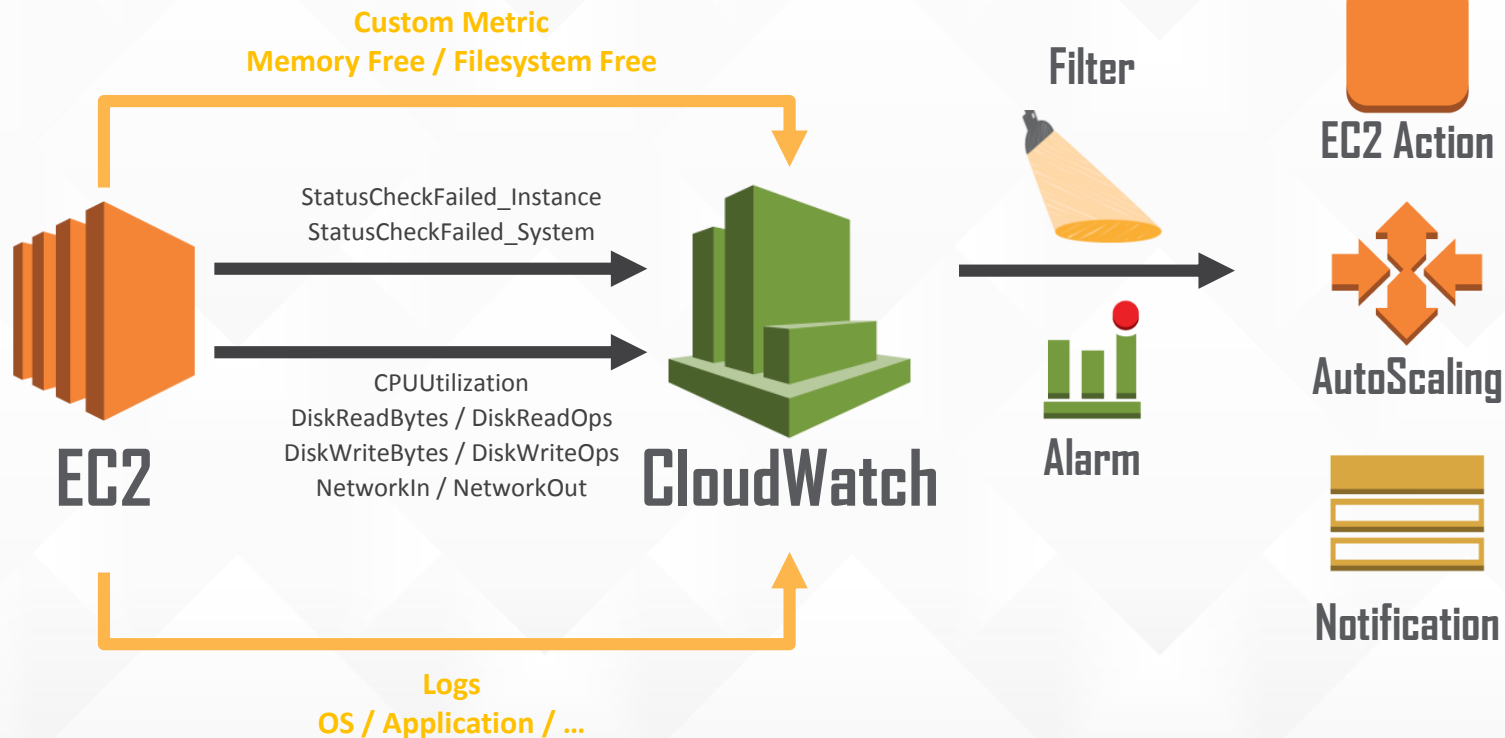
- Notes

- 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

Amazon CloudWatchのアクション機能

モニタリング

アクション



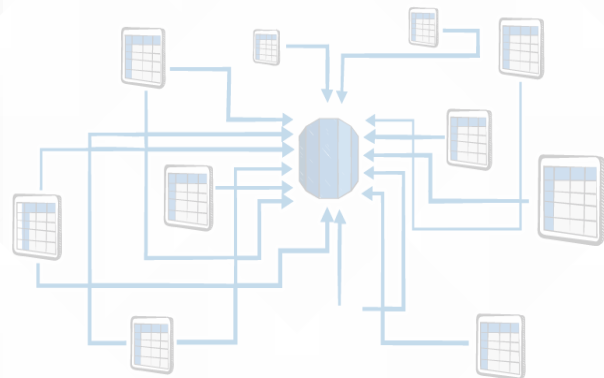
システム基盤運用で考えるポイント



Monitoring
監視

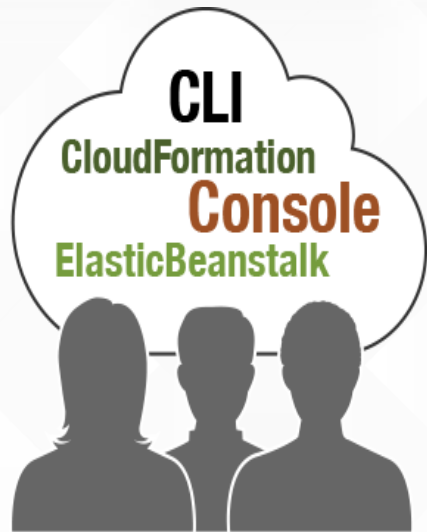


Logging
ロギング



Configuration
構成管理

AWS CloudTrail



ユーザによる
APIの発行



各リージョンの
AWSリソースの
呼び出し



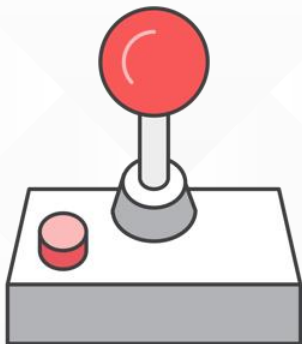
CloudTrailが
APIコールを
ロギング

| User | Action | Time |
|------|---------|--------|
| Tim | Created | 1:30pm |
| Sue | Deleted | 2:40pm |
| Kat | Created | 3:30pm |

ユーザの操作
を管理

AWS CloudTrailログの利用ケース

検知



検索



可視化



CloudWatch Logs Metric Filter の利用



- CloudTrail と CloudWatch Logs の連携
 - CloudTrailのログをJSON形式でCloudWatch Logsに転送
 - アカウント内でコールされた特定のAPIを監視し、呼ばれたときに電子メール通知を受けることが可能



CloudWatch Logs Metric Filter の利用



- 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アラーム CloudFormationテンプレート



CloudFormationをつかったメトリック フィルタの自動作成

```
1 {
2   "AWSTemplateFormatVersion" : "2010-09-09",
3   "Description" : "AWS CloudTrail API Activity Alarm Template for
CloudWatch Logs",
4   "Parameters" : {
5     "LogGroupName" : {
6       "Type" : "String",
7       "Default" : "CloudTrail/DefaultLogGroup",
8       "Description" : "Enter CloudWatch Logs log group name. Default
is CloudTrail/DefaultLogGroup"
9     },
10    "Email" : {
11      "Type" : "String",
12      "Description" : "Email address to notify when an API activity
has triggered an alarm"
13    }
14  },
15  "Resources" : {
16    "SecurityGroupChangesMetricFilter" : {
17      "Type": "AWS::Logs::MetricFilter",
18      "Properties": {
19        "LogGroup": { "Ref": "LogGroup" },
```

The screenshot shows three CloudWatch alarms created from CloudFormation templates:

- Alarm 1:** Filter Name: CloudWatchAlarm-CloudTrail-APIActivity-ConsoleSignInFailuresMetric Filter-35PVKU305SE3. Filter Pattern: { (\$eventSource = ConsoleLogin) && (\$errorMessage = "Failed authentication") }. Metric: CloudTrailMetrics / ConsoleSignInFailureCount. Metric Value: 1. Alarm: CloudTrailConsoleSignInFailures.
- Alarm 2:** Filter Name: CloudWatchAlarm-CloudTrail-APIActivity-EC2InstanceChangesMetric Filter-1Q1TJK84379GF. Filter Pattern: { (\$eventName = RunInstances) || (\$eventName = RebootInstances) || (\$eventName = StartInstances) || (\$eventName = StopInstances) || (\$eventName = TerminateInstances) }. Metric: CloudTrailMetrics / EC2InstanceEventCount. Metric Value: 1. Alarm: CloudTrailEC2InstanceChanges.
- Alarm 3:** Filter Name: CloudWatchAlarm-CloudTrail-APIActivity-EC2LargeInstanceChangesMetric Filter-r-V2DOYX90SA20. Filter Pattern: { ((\$eventName = RunInstances) || (\$eventName = RebootInstances) || (\$eventName = StartInstances) || (\$eventName = StopInstances) || (\$eventName = TerminateInstances)) && ((\$requestParameters.instanceType = *.8xlarge) || (\$requestParameters.instanceType = *.4xlarge)) }. Metric: CloudTrailMetrics / EC2LargeInstanceEventCount. Metric Value: 1. Alarm: CloudTrailEC2LargeInstanceChanges.

http://aws.typepad.com/aws_japan/2015/03/cloudtrail-integration-with-cloudwatch-in-four-more-regions.html

CloudFormationテンプレート JSONサンプル



- コンソールログインの監視

```
"ConsoleSignInFailuresMetricFilter": {
  "Type": "AWS::Logs::MetricFilter",
  "Properties": {
    "LogGroupName": { "Ref" : "LogGroupName" },
    "FilterPattern": "{ ($.eventName = ConsoleLogin) && ($.errorMessage =
Failed authentication\)" },
    "MetricTransformations": [
      {
        "MetricNamespace": "CloudTrailMetrics",
        "MetricName": "ConsoleSignInFailureCount",
        "MetricValue": "1"
      }
    ]
  }
},
```

AWS CloudTrail API Activity Lookup



Filter: **Event name** Enter event name Time range: Select time range

| Event time | Event name | Resource type | Resource name |
|-------------------------|-----------------------------|-------------------|---------------------------------|
| 2015-05-16, 12:29:19 PM | CreateCluster | Subnet and 3 more | subnet-0313056c and 4 more |
| 2015-05-16, 12:29:19 PM | CreateClusterParameterGroup | Subnet and 3 more | subnet-0513056c and 4 more |
| 2015-05-16, 12:29:18 PM | CreateClusterSecurityGroup | Subnet and 3 more | subnet-0913056c and 4 more |
| 2015-05-16, 12:29:18 PM | CreateClusterSnapshot | Subnet and 3 more | subnet-0913056c and 4 more |
| 2015-05-16, 12:29:18 PM | CreateClusterSubnetGroup | Subnet and 3 more | subnet-0913056c and 4 more |
| 2015-05-16, 12:29:14 PM | CreateConfigurationTemplate | User and 1 more | redshift-2015-05-16-12:29:14 PM |
| 2015-05-16, 12:29:11 PM | CreateConnection | User and 1 more | redshift-2015-05-16-12:29:11 PM |
| 2015-05-16, 12:29:11 PM | CreateCustomerGateway | User and 1 more | redshift-2015-05-16-12:29:11 PM |
| 2015-05-16, 12:29:10 PM | CreateDBCluster | User | redshift-2015-05-16-12:29:10 PM |
| 2015-05-16, 12:29:10 PM | CreateDBClusterSnapshot | User | redshift-2015-05-16-12:29:10 PM |
| 2015-05-16, 12:29:10 PM | CreateDBInstance | User | redshift-2015-05-16-12:29:10 PM |
| 2015-05-16, 12:29:10 PM | CreateCluster | | |

AWS access key AKIAJ77COT71184UC6G
AWS region us-west-2
Error code
Event ID c84aa2ff-38ba-4b6e-bef9-548e672bda2f
Event name CreateCluster

Event source redshift.amazonaws.com
Event time 2015-05-16, 12:29:10 PM
Request ID 270c0c1b7-445c-4c3e-6702e
Source IP address cloudformation.amazonaws.com
User name s3k3k3k3

Resources Referenced (0)

[View event](#)

```
$ aws cloudtrail lookup-events --lookup-attributes  
AttributeKey=EventName,AttributeValue=RunInstances --start-time 2015-05-16  
--end-time 2015-05-16
```

Amazon CloudSearch, Amazon Lambda



Search: event_source:'sqs.amazonaws.com' **Go** Options No Suggester ▾

Filter Query:

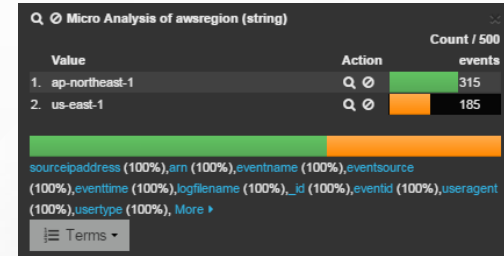
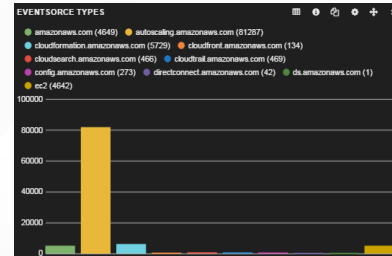
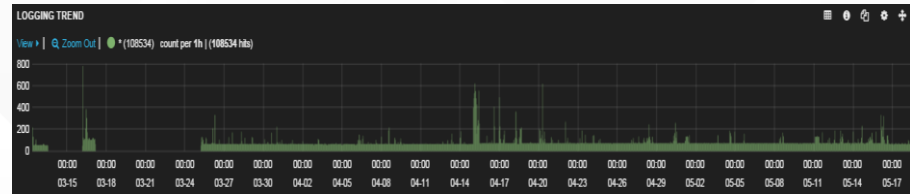
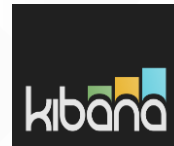
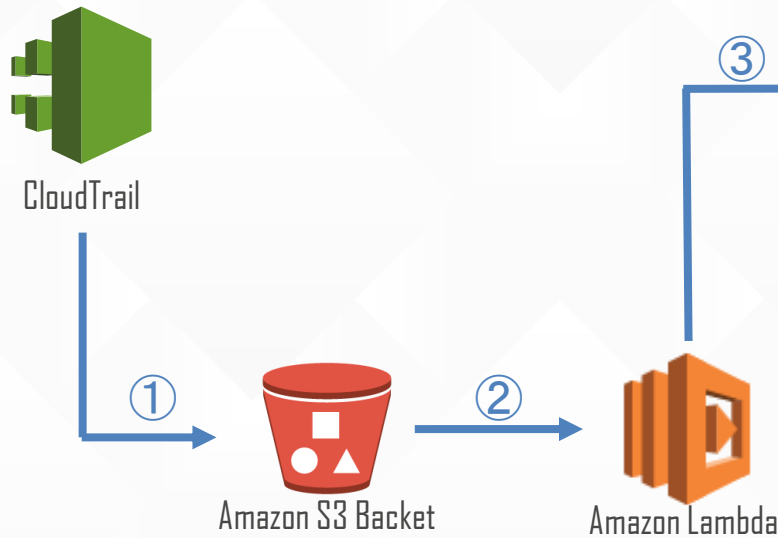
Query Parser: Structured ▾ Default Operator: and (default) ▾ ⓘ

Search Fields:

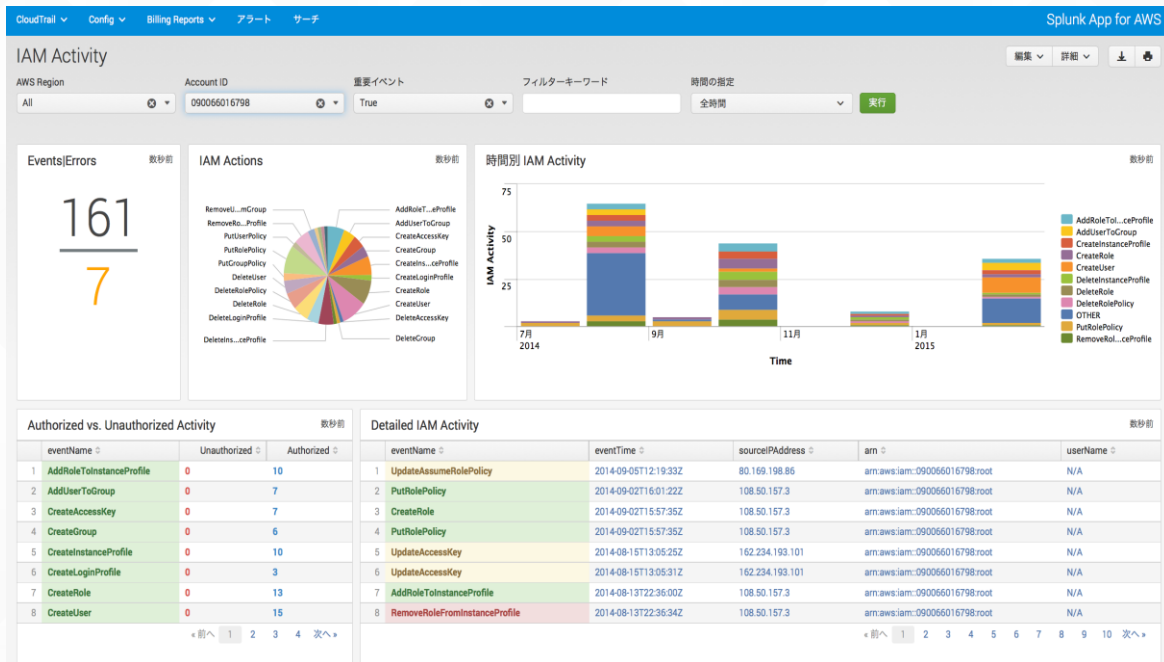
Sort by: Document Score ▾ Descending ▾ (view raw: JSON or XML) ⏪ 1 to 10 of 23 Results ⏩

| | | |
|---|---|-------------------------|
| 1. 325562c8-a8cd-42d8-b0b2-a2194f89686a | | Filter Search Results |
| _score | 10.123007 | aws_region |
| user_identity_arn | arn:aws:iam::926983381032:user/root | eu-west-1 (23) |
| event_source | sqs.amazonaws.com | error_code |
| event_time | 2014-09-22T12:41:00Z | event_name |
| source_ip_address | 54.240.197.233 | SetQueueAttributes (13) |
| event_id | 325562c8-a8cd-42d8-b0b2-a2194f89686a | CreateQueue (7) |
| aws_region | eu-west-1 | DeleteQueue (3) |
| user_identity_user_name | root | source_ip_address |
| raw | { "eventVersion": "1.01", "eventID": "325562c8-a8cd-42d8-b0b2-a2194f89686a", "eventTime": "... more..." | 54.240.197.233 (6) |
| user_agent | signin.amazonaws.com | 80.12.59.193 (5) |
| user_identity_account_id | 926983381032 | 54.240.197.234 (4) |
| user_identity_type | IAMUser | |
| event_name | SetQueueAttributes | |
| 2. b3836873-731d-4a1e-bc8e-47ba690ab7c4 | | |
| _score | 10.123007 | |

ElasticSearch, Kibana, Amazon Lambda



Splunk App for AWS



```
新規サーチ
host="ip-10-170-32-235" sourcetype="aws:cloudtrail" eventName=DeleteInstance*
```

```
時間 イベント
> 15/02/03 { [-]
15:18:26.000
awsRegion: us-east-1
eventName: DeleteInstanceProfile
eventSource: iam.amazonaws.com
eventTime: 2015-02-03T15:18:26Z
eventType: AwsApiCall
eventVersion: 1.02
recipientAccountId: 090066016798
requestID: e6654a76-abb7-11e4-b773-95f72fc3d47
requestParameters: { [+]}
responseElements: null
sourceIPAddress: 173.63.70.227
userAgent: console.amazonaws.com
userIdentity: { [+]}
}
```



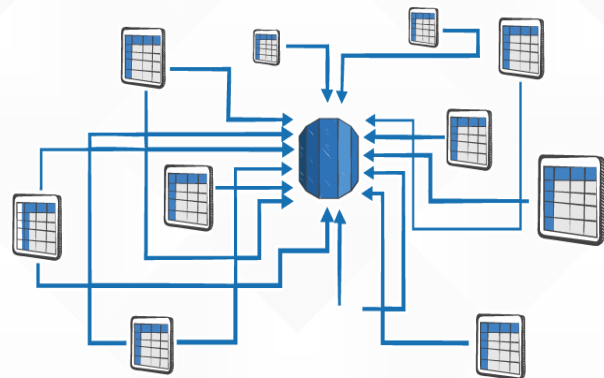
システム基盤運用で考えるポイント



Monitoring
監視

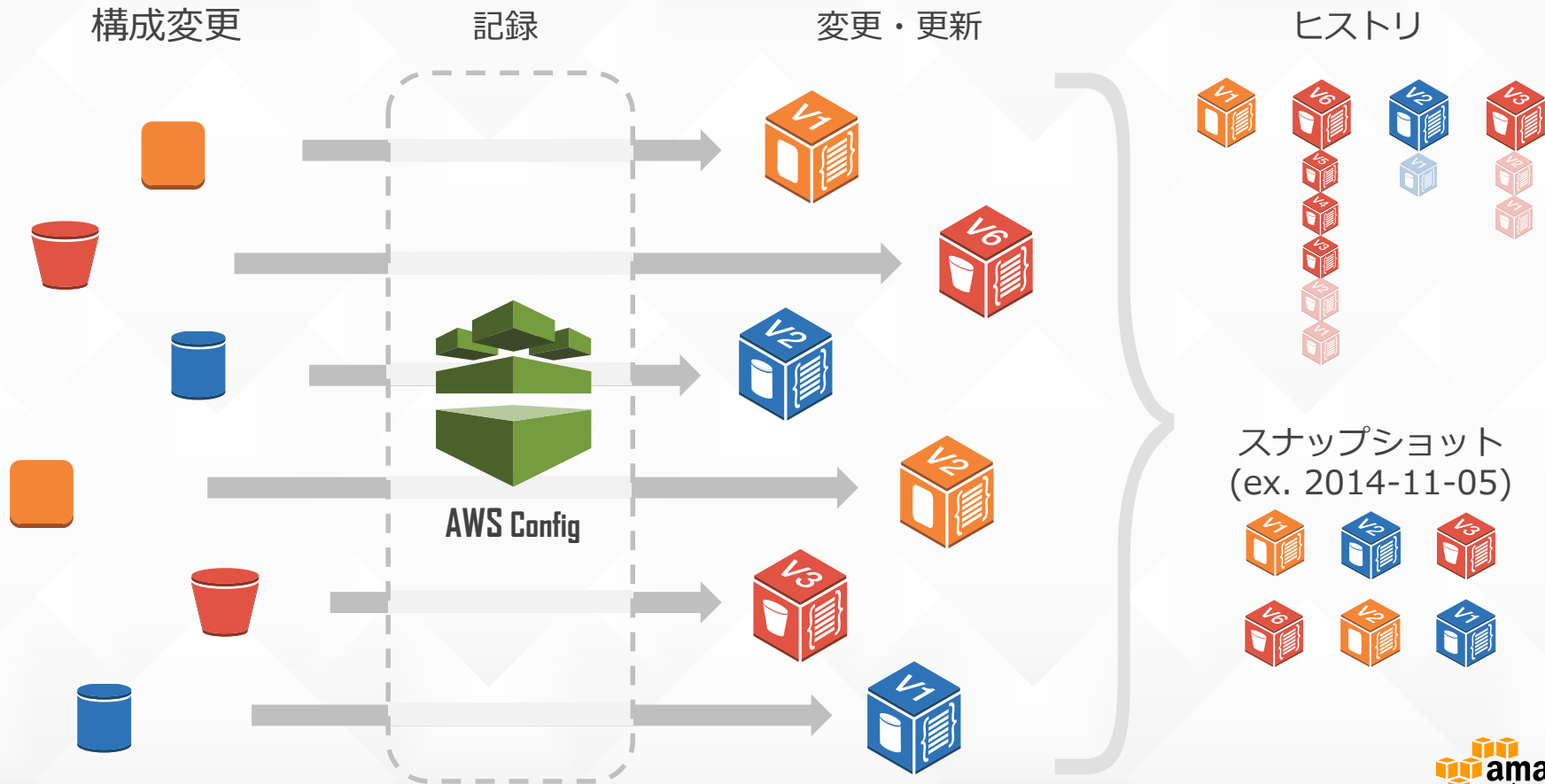


Logging
ロギング



Configuration
構成管理

AWS Config

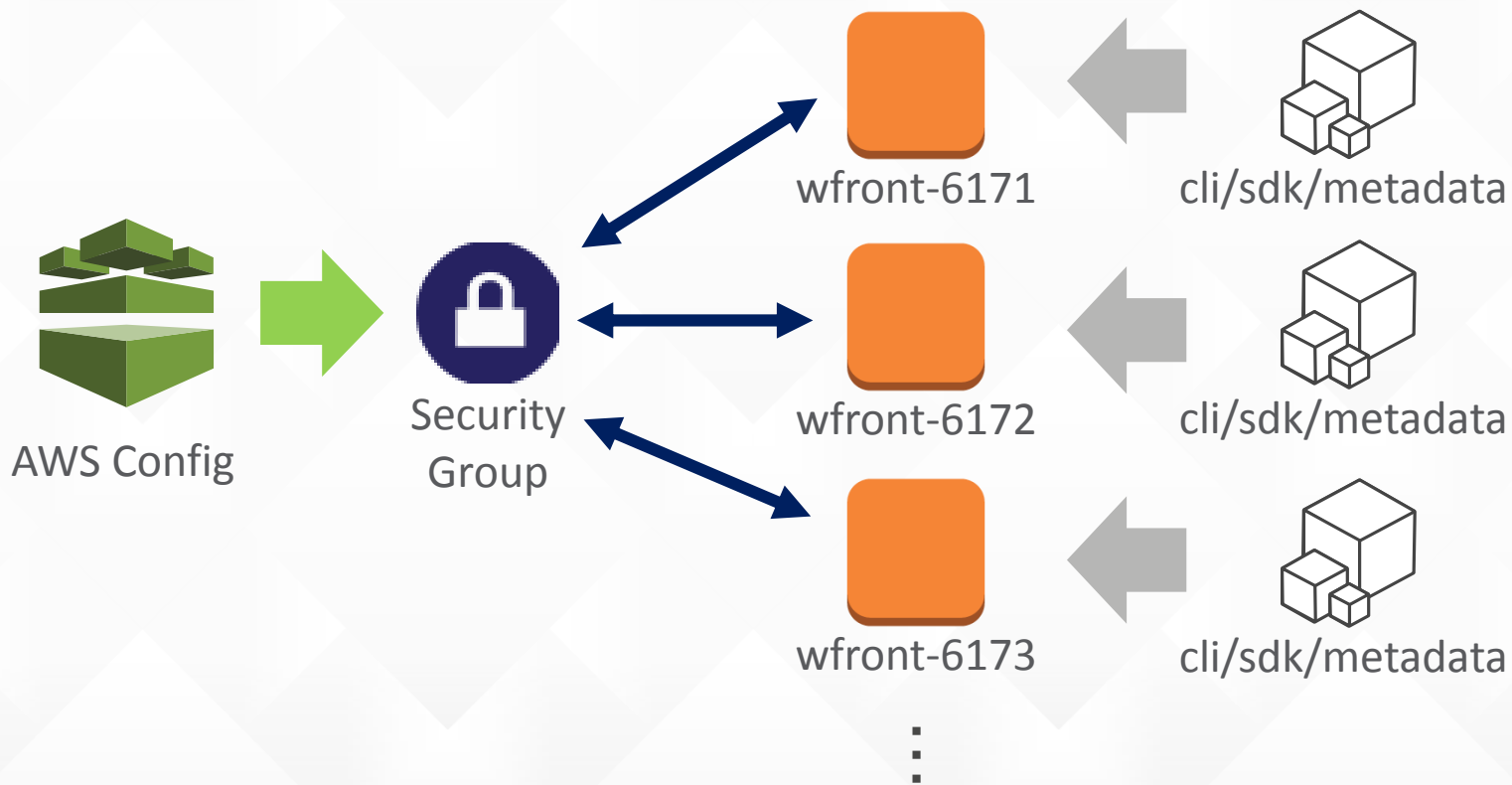


AWS Configから見るリソースのリレーション

```
$ aws configservice get-resource-config-history --resource-type
AWS::EC2::Instance --resource-id i-cbe78721 --region us-east-1
{
  "configurationItems": [
    {
      "configurationItemCaptureTime": 1427432204.542,
      "resourceCreationTime": 1417001136.0,
      "availabilityZone": "us-east-1c",
      "tags": {
        "Name": "cwl-sqlserver"
      },
      "resourceType": "AWS::EC2::Instance",
      "resourceId": "i-cbe78721",
      "configurationStateId": "547",
      "relatedEvents": [],
      "relationships": [
        {
          "resourceType": "AWS::EC2::EIP",
          "resourceId": "eipalloc-e6dc7283",
          "relationshipName": "Is attached to ElasticIp"
        }
      ],
    }
  ],
}
```

```
{
  "resourceType": "AWS::EC2::NetworkInterface",
  "resourceId": "eni-827041f4",
  "relationshipName": "Contains NetworkInterface"
},
{
  "resourceType": "AWS::EC2::SecurityGroup",
  "resourceId": "sg-53f98236",
  "relationshipName": "Is associated with SecurityGroup"
},
{
  "resourceType": "AWS::EC2::Subnet",
  "resourceId": "subnet-a1db1cd6",
  "relationshipName": "Is contained in Subnet"
},
{
  "resourceType": "AWS::EC2::Volume",
  "resourceId": "vol-592a8313",
  "relationshipName": "Is attached to Volume"
},
  ...
}
```

AWS Config リレーションシップ



AWS Configからのスナップショットの取得

```
$ aws configservice deliver-config-snapshot --delivery-channel-name default --region us-east-1
{
  "configSnapshotId": "5b662925-a6e0-4d19-a156-0151d932178f"
}
$ aws configservice describe-delivery-channel-status --region us-east-1
{
  "DeliveryChannelsStatus": [
    {
      "configStreamDeliveryInfo": {
        "lastStatusChangeTime": 1431767053.909,
        "lastStatus": "SUCCESS"
      },
      "configHistoryDeliveryInfo": {
        "lastSuccessfulTime": 1431699583.504,
        "lastStatus": "SUCCESS",
        "lastAttemptTime": 1431699583.504
      },
      "configSnapshotDeliveryInfo": {
        "lastSuccessfulTime": 1431769523.567,
        "lastStatus": "SUCCESS",
        "lastAttemptTime": 1431769523.567
      },
      "name": "default"
    }
  ]
}
```

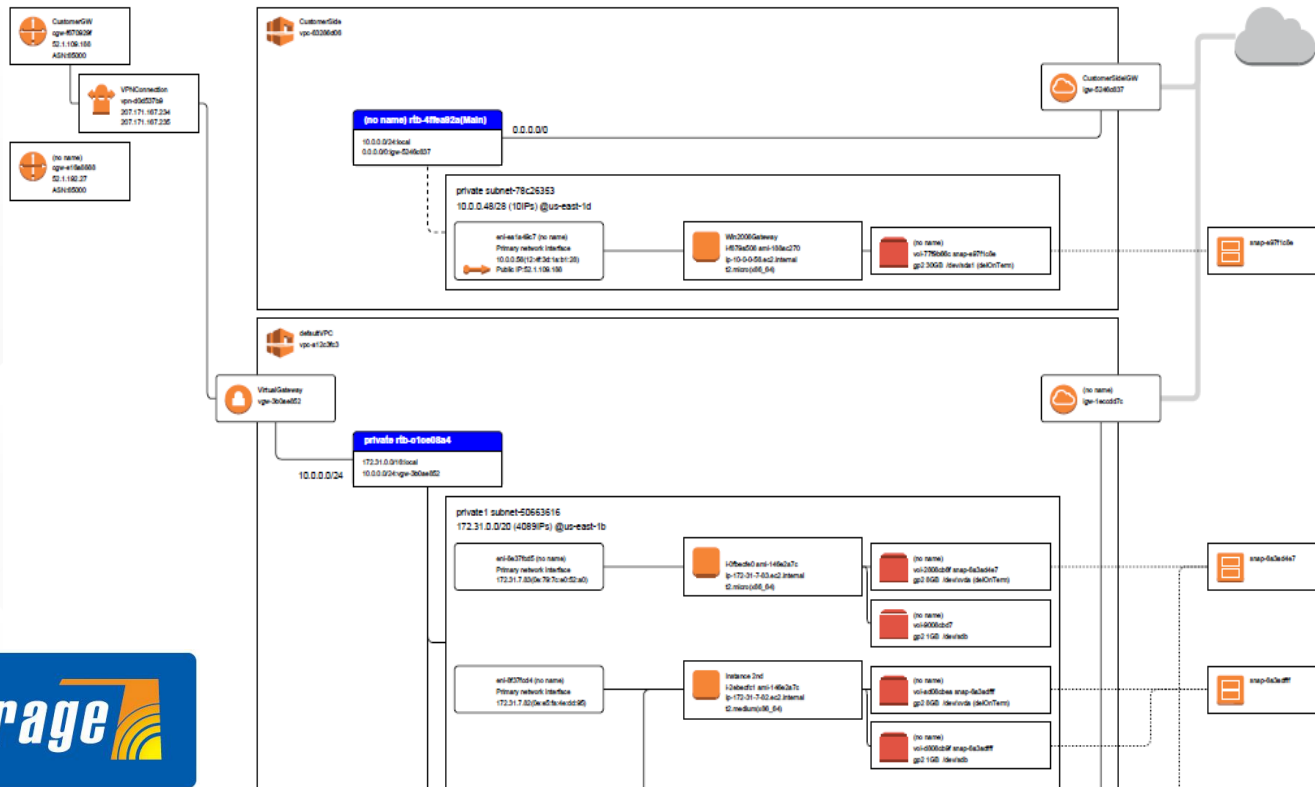
Logstorageによる可視化



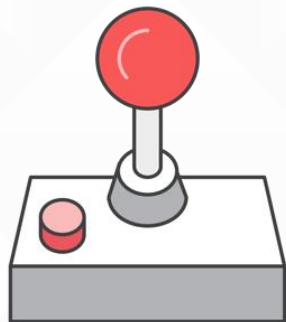
AWS Config snapshot Report

Snapshot created: 2015/05/19 10:25:08 +0900

data2



まとめ



モニタリング



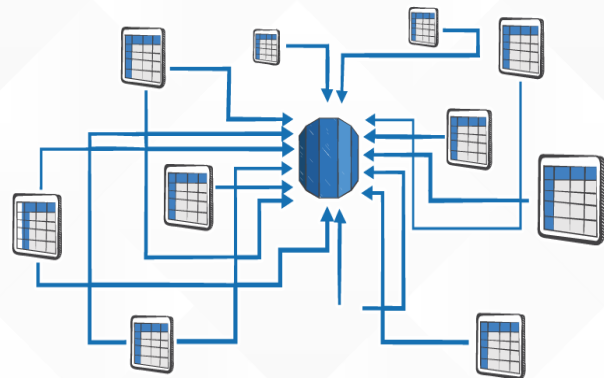
Amazon CloudWatch



インフラ監査



AWS CloudTrail



構成管理



AWS Config

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

ESP for AWS - ソフトウェア / SaaSカタログ

ESP

Ecosystem Solution Pattern 2015

2015年度版



AWS対応ソフトウェア / SaaSカタログ

amazon Partner Network

JPI
(商用転売)ショップ
株式会社 日立製作所
統合システム運用管理 詳細
システム構成図:

対話ユーザー導入価値:
JPI Verionetは、AWS環境、オンプレシステムを維持するにあたり、サービスの向上とコスト削減に貢献しています。
製品 / ソリューション提供の強み:
1994年の創業を期したJPIは、システム運用管理領域で長年のノウハウを蓄積し、企業全体のシステム運用管理の課題、ノウハウを幅広く提供し、お客様のシステム運用を積極的に支援します。JPIはサービス「バージョン」のリリースからお客様の要望や、システム運用管理の課題を察知し、様々な場面に合わせた多数の課題を解決する為、迅速な課題解決を実現する製品です。そして、最終的にお客様のシステム運用管理を定常化するため、10年以上の稼働実績のある保守契約（O&M）サービスを提供している業界唯一のAmazon EC2に最適化されたAWS環境では、AWSの「オースタークル」で実際の稼働を維持しています。特に、Amazon CloudWatchの活用により、定期的な監視からリアルタイムでの監視に切り替わり、

Amazon Data Services Japan K.K.

システム提供/提供するAWS:
■対応 OS: Windows Server(R3) / Windows Server®2003 R2 / Windows Server®2008 / Windows Server®2012 R2 / RHEL、など
■製品提供先: amazon-ec2、amazon-ecs、amazon-elasticbeanstalk

システム提供/提供するAWS:
■対応 OS: Windows Server(R3) / Windows Server®2003 R2 / Windows Server®2008 / Windows Server®2012 R2 / RHEL、など
■製品提供先: amazon-ec2、amazon-ecs、amazon-elasticbeanstalk

詳細: <http://www.hinemo.com/jp/freelife>

システム提供/提供するAWS:
■対応 OS: Windows Server(R3) / Windows Server®2003 R2 / Windows Server®2008 / Windows Server®2012 R2 / RHEL、など
■製品提供先: amazon-ec2、amazon-ecs、amazon-elasticbeanstalk

PLAT
(商用転売)ショップ
株式会社 野村総合研究所
PLAT (エムプラットフォーム)
システム構成図:

対話ユーザー導入価値:
JPI Verionetは、AWS環境、オンプレシステムを維持するにあたり、サービスの向上とコスト削減に貢献しています。
製品 / ソリューション提供の強み:
1994年の創業を期したJPIは、システム運用管理領域で長年のノウハウを蓄積し、企業全体のシステム運用管理の課題、ノウハウを幅広く提供し、お客様のシステム運用を積極的に支援します。JPIはサービス「バージョン」のリリースからお客様の要望や、システム運用管理の課題を察知し、様々な場面に合わせた多数の課題を解決する為、迅速な課題解決を実現する製品です。そして、最終的にお客様のシステム運用管理を定常化するため、10年以上の稼働実績のある保守契約（O&M）サービスを提供している業界唯一のAmazon EC2に最適化されたAWS環境では、AWSの「オースタークル」で実際の稼働を維持しています。特に、Amazon CloudWatchの活用により、定期的な監視からリアルタイムでの監視に切り替わり、

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



Thank You