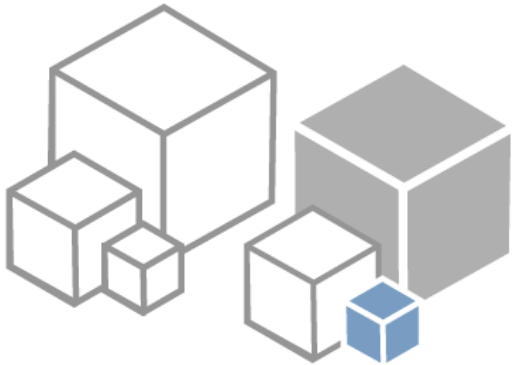
AWS Command Line Interface & AWS Tools for Windows PowerShell

2015/07/22
AWS Black Belt Tech Webinar 2015

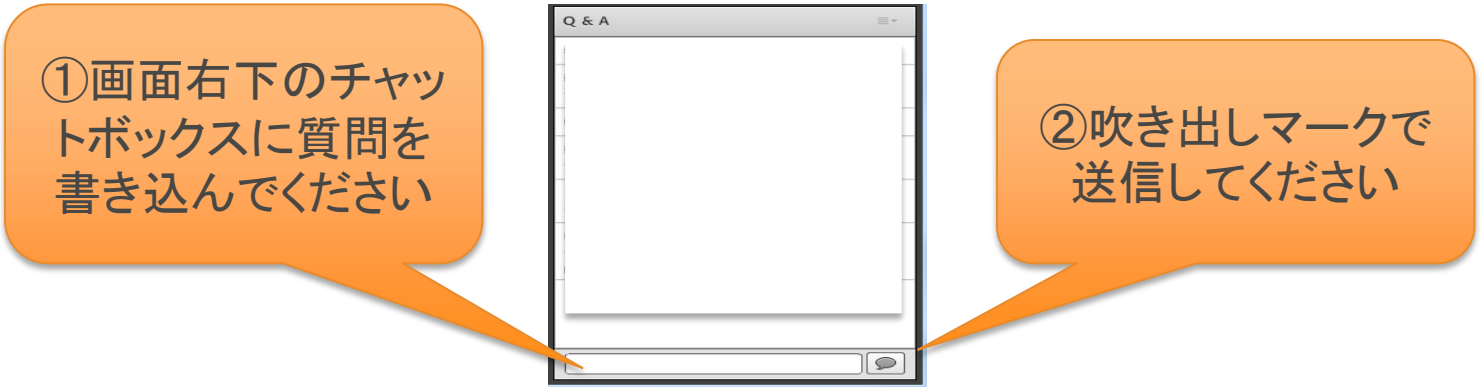
アマゾンデータサービスジャパン株式会社

プロフェッショナルサービス 千葉悠貴



AWS Black Belt Tech Webinar へようこそ！

- 質問を投げることができます！
  - Adobe Connectのチャット機能を使って、質問を書き込んでください。（書き込んだ質問は、主催者にしか見えません）
  - Twitterへツイートする際はハッシュタグ**#awsblackbelt**をご利用ください。

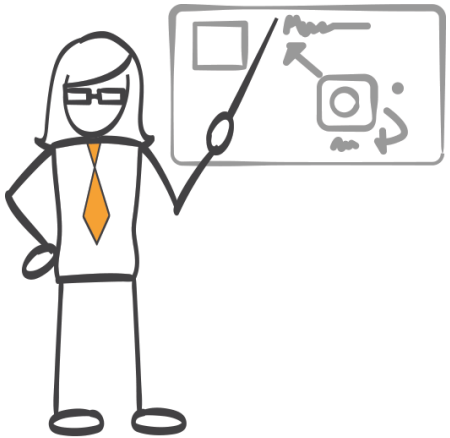


AWS Black Belt Tech Webinar 2015

- 今後の配信予定

  7月は「**AWS 運用機能月間**」、８月は「**DB月間**」です！

  - 7/29（水）AWS CloudHSM & AWS KMS
  - 8/5 （水）Amazon DynamoDB
  - 8/12（水）お盆のためお休み
  - 8/19（水）Amazon ElastiCache
  - 8/26（水）Amazon Redshift



- イベントスケジュール

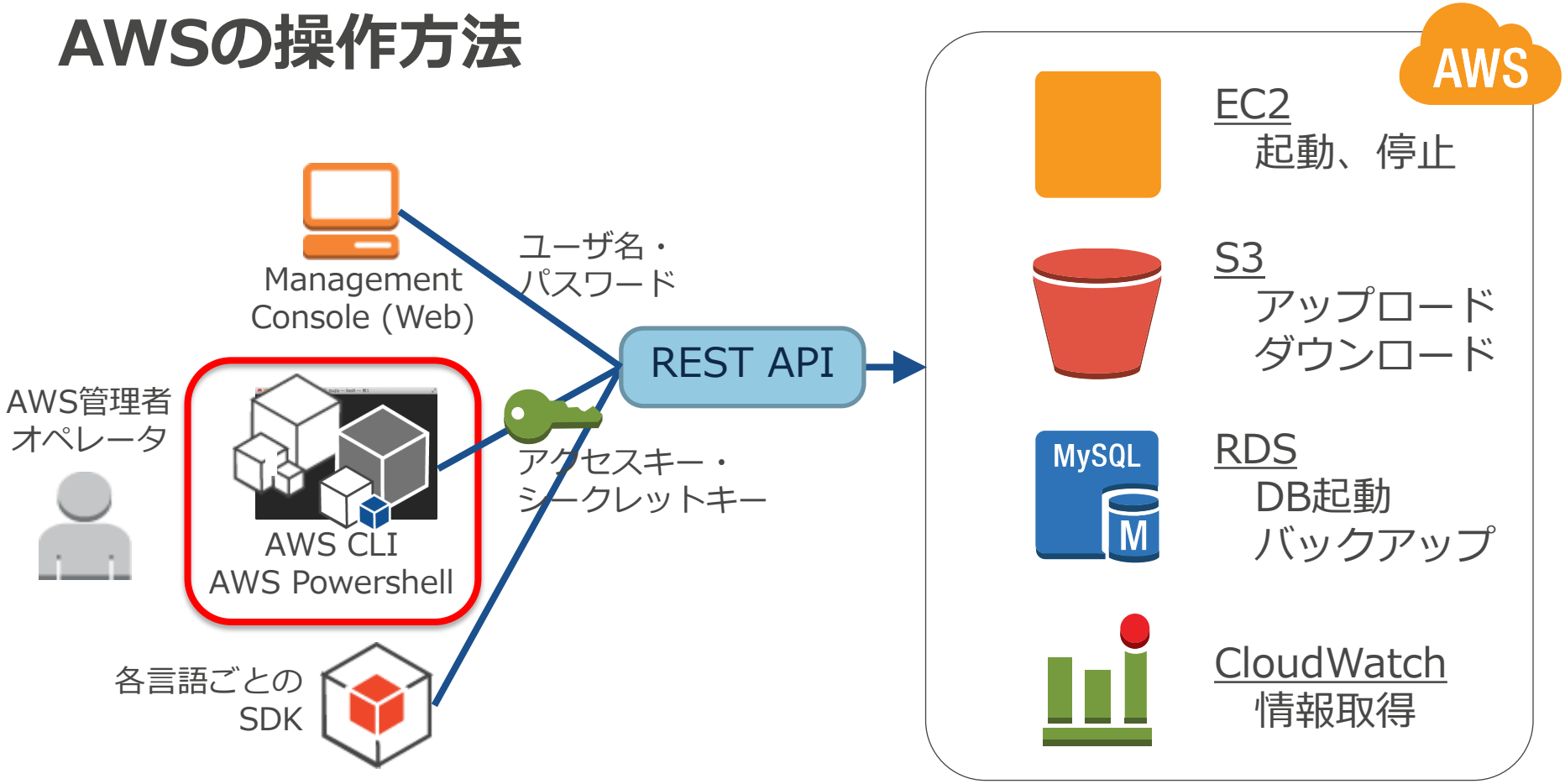
  http://aws.amazon.com/jp/event_schedule/

アジェンダ

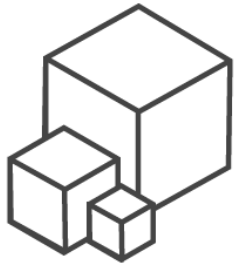
- AWS CLI / Powershellの概要
- AWS CLI / Powershell のセットアップ
- AWS Command Line Interface
- AWS Tools for Windows PowerShell

AWS CLI / Powershell の概要

AWSの操作方法


Management
Console (Web)
ユーザ名・
パスワード
REST API
AWS管理者
オペレータ
AWS CLI
AWS Powershell
アクセスキー・
シークレットキー
各言語ごとの
SDK
AWS
EC2
起動、停止
S3
アップロード
ダウンロード
MySQL
RDS
DB起動
バックアップ
CloudWatch
情報取得

AWS Command Line Interface (CLI)



- “aws”という名前の単一コマンドでAWSサービスを操作可能
- プラットフォームや開発言語などが限定されない
  Windows, Linux, Mac, Unixなど
- S3用にはsyncなどの便利な機能あり

AWS Tools for Windows PowerShell

- “AWSPowerShell”モジュール内のコマンドレットから、ほとんどのAWSサービスを操作可能
- PowerShellの強力なシェル機能が利用できる

対応サービス①

(2015年7月時点)

サービス名	CLI	Power Shell
Amazon CloudFront	○	○
Amazon CloudHSM	○	○
Amazon CloudSearch	○	○
Amazon CloudSearch Domain	○	○
Amazon CloudWatch	○	○
Amazon CloudWatch Logs	○	○
Amazon Cognito Identity	○	-
Amazon Cognito Sync	○	-
Amazon DynamoDB	○	○
Amazon DynamoDB Streams	○	○
Amazon EC2 Container Service	○	○
Amazon Elastic Compute Cloud	○	○
Amazon Elastic File System	○	○

サービス名	CLI	Power Shell
Amazon Elastic MapReduce	○	○
Amazon Elastic Transcoder	○	○
Amazon ElastiCache	○	○
Amazon Glacier	○	-
Amazon Kinesis	○	○
Amazon Machine Learning	○	○
Amazon Redshift	○	○
Amazon Relational Database Service	○	○
Amazon Route 53	○	○
Amazon Route 53 Domains	○	○
Amazon Simple Email Service	○	○
Amazon Simple Notification Service	○	○
Amazon Simple Queue Service	○	○

対応サービス②

(2015年7月時点)

サービス名	CLI	Power Shell
Amazon Simple Storage Service	○	○
Amazon Simple Systems Management Service	○	○
Amazon Simple Workflow Service	○	-
Amazon SimpleDB	○	-
Amazon WorkSpaces	○	○
Auto Scaling	○	○
AWS CloudFormation	○	○
AWS CloudTrail	○	○
AWS CodeCommit	○	○
AWS CodeDeploy	○	○
AWS CodePipeline	○	○
AWS Config	○	○
AWS Data Pipeline	○	○

サービス名	CLI	Power Shell
AWS Device Farm	○	○
AWS Direct Connect	○	○
AWS Directory Service	○	○
AWS Elastic Beanstalk	○	○
AWS Identity and Access Management	○	○
AWS Import/Export	○	○
AWS Key Management Service	○	○
AWS Lambda	○	○
AWS OpsWorks	○	○
AWS Security Token Service	○	○
AWS Storage Gateway	○	○
AWS Support	○	○
Elastic Load Balancing	○	○

AWS CLI / Powershell のセットアップ

Step1. IAMアカウントの準備

AWS CLI／Powershellを利用するには、IAMユーザやIAMロールが必要

- IAMユーザ・IAMロールを用意する

  オンプレミス上のPCやサーバから利用
  　IAMユーザを作成し、アクセスキー・シークレットキーを発行

  EC2インスタンス上から利用
  　IAMロールを作成し、割り当てたEC2インスタンスの作成を推奨
  　IAMユーザも利用可

- IAMユーザ・IAMロールに必要な権限をIAM Policyで付与する

http://docs.aws.amazon.com/ja_jp/cli/latest/userguide/cli-chap-getting-set-up.html

Step2. インストール

- AWS CLI

プラットフォーム	インストール方法
Windows	・MSI形式インストーラ
Amazon Linux	・インストール済み
共通	・pip（pythonのパッケージ管理システム） ・バンドルインストーラ ・手動インストール

- AWS Powershell

プラットフォーム	インストール方法
Windows	・MSI形式インストーラ ・Powershell Gallery (Powershellのパッケージ管理システム)
Windows on EC2	・インストール済み

http://docs.aws.amazon.com/ja_jp/cli/latest/userguide/installing.html
http://docs.aws.amazon.com/ja_jp/powershell/latest/userguide/pstools-getting-set-up.html

Powershell Galleryサポート

2015年5月

Powershell Gallery

- Microsoftが提供するPowershellの統合リポジトリ（2015年7月時点ではプレビュー中）
- リポジトリから最新モジュールのインストール／アップデートが可能
- 利用開始前にWindows Management Framework v5 previewのインストールが必要

- AWS Powershellのインストール

```
PS C:¥> Install-Module -Name AWSPowerShell
```

- AWS Powershellのアップデート

```
PS C:¥> Update-Module -Name AWSPowerShell
```

※アップデートは、 Powershell Galleryを利用してインストールしたモジュールのみ実施可能です。
AWS Powershellが既にインストールされている場合、一旦アンインストールしてから再インストールする必要があります。

https://blogs.aws.amazon.com/net/post/Tx3E4TIMWDG4604/Announcing-Support-for-the-PowerShell-Gallery
https://www.powershellgallery.com/packages/AWSPowerShell/

Step3. 初期設定

AWSクレデンシャル、リージョン、出力形式などを設定

- AWS CLI

  ”aws configure”で設定

```
$ aws configure
AWS Access Key ID [None]: XXXXXXXXXXX
AWS Secret Access Key [None]: XXXXXXXXXXXXXXXXXXXXXXX
Default region name [None]: ap-northeast-1
Default output format [None]:json
```

- AWS Powershell

  ”Initialize-AWSDefaults”で設定

```
PS C:¥> Initialize-AWSDefaults -AccessKey XXXXXXX -SecretKey XXXXXXXXX -Region ap-northeast-1
```

  Powershellオブジェクト形式で出力されるため、出力形式の指定は不要

http://docs.aws.amazon.com/ja_jp/cli/latest/userguide/cli-chap-getting-started.html

http://docs.aws.amazon.com/ja_jp/powershell/latest/userguide/specifying-your-aws-credentials.html

プレビュー機能の有効化 (AWS CLI)

一部のサービスのAWS CLIはプレビュー版です。
プレビュー機能を利用する場合、aws configureで有効化する必要があります。

2015年7月時点プレビュー機能（バージョン1.7.39）

- Amazon CloudFront -- aws cloudfront
- Amazon Elastic File System -- aws efs
- Amazon SimpleDB -- aws sdb

設定方法

```
$ aws configure set preview.cloudfront true
$ aws configure set preview.efs true
$ aws configure set preview.sdb true
```

http://docs.aws.amazon.com/cli/latest/reference/configure/set.html

aws s3コマンド設定 (AWS CLI)

2015年2月

aws configureコマンド（または~/.aws/configファイル）で、
aws s3コマンドの設定項目が追加されました。

設定項目	内容	デフォルト値
max_concurrent_requests	S3へのリクエストの最大並列数	10
max_queue_size	S3のタスクキューの最大サイズ	1000
multipart_threshold	ファイルをマルチパートに分割するしきい値	8MB
multipart_chunksize	マルチパートに分割する際のチャンクサイズ	8MB

設定方法

```
$ aws configure set default.s3.max_concurrent_requests 20
$ aws configure set default.s3.max_queue_size 10000
$ aws configure set default.s3.multipart_threshold 64MB
$ aws configure set default.s3.multipart_chunksize 16MB
```

http://docs.aws.amazon.com/cli/latest/topic/s3-config.html

複数プロファイルの利用

設定内容はプロファイルという単位で保存し、実行時に指定可能

利用例：開発環境と本番環境でクレデンシャルが分かれている場合など

- AWS CLI

  ”aws configure”で保存、 ”--profile”で指定

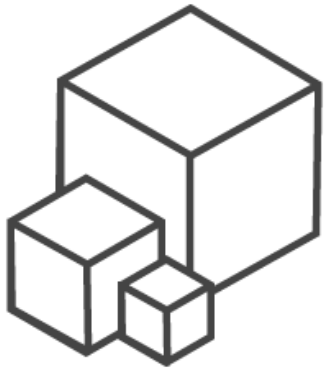
```
$ aws configure --profile my-profile-name
$ aws ec2 describe-instances --profile my-profile-name
```

- AWS Powershell

  ”Set-AWSCredentials”で保存、 ”-ProfileName”で指定

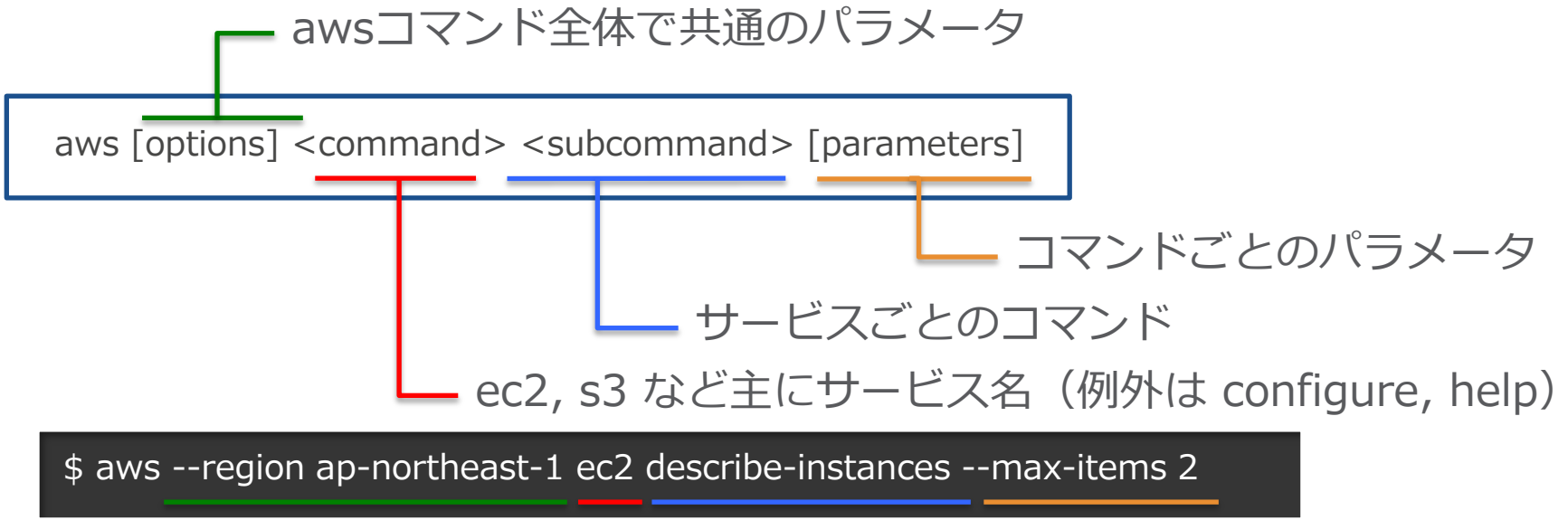
```
PS C:¥> Set-AWSCredentials -AccessKey XXXXXXXX -SecretKey XXXXXXXXX -StoreAs MyProfileName
PS C:¥> Get-EC2Instance -ProfileName MyProfileName
```

AWS Command Line Interface



パラメータの形式

AWS CLIコマンドのパラメータ形式は以下の通りです。
全ての[options]、[parameters]は -- から始まります。



command Reference: http://docs.aws.amazon.com/cli/latest/reference/

リターンコード

AWS CLIを実行した結果のリターンコードには以下の種類があります。

リターンコード	意味
0	エラー無しでコマンド実行が成功
1	s3コマンドで制限値に到達。最低１つのs3転送が失敗
2	パラメータ不足などのエラー。またはs3コマンドで一部ファイルのみ転送失敗。
255	コマンド実行失敗。CLIまたはサービス側のエラー。

確認方法

```
$ aws ec2 describe-instances
$ echo $?
0 #成功時
```

http://docs.aws.amazon.com/cli/latest/topic/return-codes.html

[options]： --region

コマンドを実行するリージョンを指定します。設定ファイルで指定されたリージョン設定より優先されます。

```
aws --region (name) [options] <command> <subcommand> [parameters]
```

ap-northeast-1	アジアパシフィック（東京）リージョン
ap-southeast-1	アジアパシフィック（シンガポール）リージョン
ap-southeast-2	アジアパシフィック（シドニー）リージョン
eu-central-1	欧州（フランクフルト）リージョン
eu-west-1	欧州（アイルランド）リージョン
sa-east-1	南米（サンパウロ）リージョン
us-east-1	米国東部（バージニア北部）リージョン
us-west-1	米国西部（北カリフォルニア）リージョン
us-west-2	米国西部（オレゴン）リージョン

[options]： --output

コマンド出力のフォーマットを指定します。設定ファイルで指定されたフォーマット設定より優先されます。デフォルトはJSON形式です。

```
aws --output (name) [options] <command> <subcommand> [parameters]
```

json	JSON形式
text	テキスト形式
table	テーブル形式

[options] : --output

JSON

```
{
   "Places": [
       {
            "City": "Seattle",
            "State": "WA"
       },
       {
            "City": ”Las Vegas",
            "State": "NV"
       }
   ]
}
```

Text

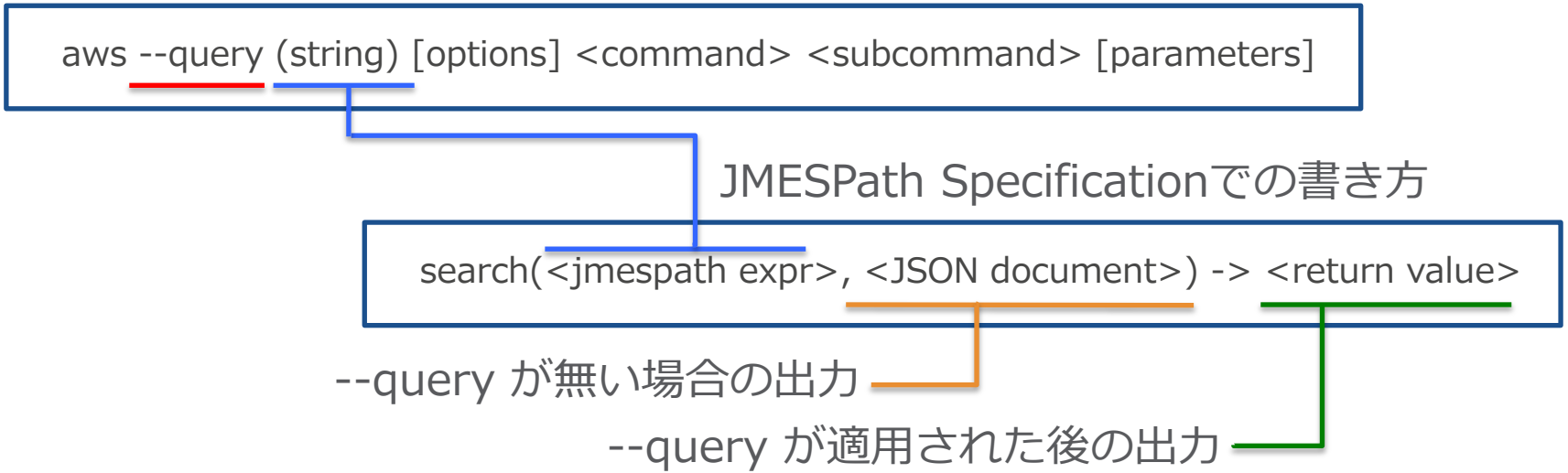
```
PLACES  Seattle    WA
PLACES  Las Vegas  NV
```

Table

```
-------------------------
|     SomeOperationName  |
+------------------------+
||         Places       ||
|+------------+---------+|
|| City       | State   ||
|+------------+---------+|
|| Seattle    | WA      ||
|| Las Vegas  | NV      ||
|+------------+---------+|
```

[options]： --query

awsコマンドの出力を、JSONの構造に基いてJMESPathクエリーでフィルタリング等ができる強力な仕組みです。



http://docs.aws.amazon.com/ja_jp/cli/latest/userguide/controlling-output.html#controlling-output-filter

JMESPath

JMESPathはJSONのためのクエリー言語です。
JSONのパース、フィルター、整形などができます。

JMESPath.org

http://jmespath.org

チュートリアル

http://jmespath.org/tutorial.html

http://jmespath.org/examples.html

JMESPathクエリーの基本的な表記方法

クエリー種別	インプット	クエリー	結果
キー	{"a": "foo", "b": "bar"}	a	foo

キー(階層)	{"a": {
	"b": {
	"c":"value"
	}
	}
	}
		a.b.c	value

インデックス	["a", "b", "c", "d", "e", "f"]	[1]	b
スライス	[0, 1, 2, 3, 4, 5]	[0:4]	[0, 1, 2, 3]
スライス(スキップ)	[0, 1, 2, 3, 4, 5]	[::2]	[0, 2, 4]
パイプ	[0, 1, 2, 3, 4, 5]	[::2] | [1]	2

JMESPathクエリーの基本的な表記方法

クエリー種別	インプット	クエリー	結果
==	{"machines": [{"name": "a", "state": "running" }, {"name": "b", "state": "stopped" }]}	machines[?state== 'running'].name	["a"]
!=	{"machines": [{"name": "a", "state": "running" }, {"name": "b", "state": "stopped" }]}	machines[?state!= 'running'].name	["b"]

マルチセレクト	{"machines": [{
	"name": "a",
	"state": {"name": "running"}
	},{
	"name": "b",
	"state": {"name": "stopped"}
	}]}
		machines[].[name,
		state.name]
			[
			["a","running"],
			["b","stopped"]
			]

JMESPathクエリーの基本的な表記方法

```
{
 "Users": [
    {
     "Arn": "arn:aws:iam::XXXX:user/james",
     "UserId": "userid",
     "CreateDate": "2013-03-09T23:36:32Z",
     "Path": "/",
     "UserName": "james"
    }
  ]
}
```

JMESPathクエリーの基本的な表記方法

```
--query Users[0].[UserName,Path,UserId]
  {
   "Users": [
      {
       "Arn": "arn:aws:iam::XXXX:user/james",
       "UserId": "userid",
       "CreateDate": "2013-03-09T23:36:32Z",
       "Path": "/",
       "UserName": "james"
      }
    ]
  }
```

JMESPathクエリーの基本的な表記方法

```
--query Users[0].[UserName,Path,UserId]
```

```
{
 "Users": [
    {
     "Arn": "arn:aws:iam::XXXX:user/james",
     "UserId": "userid",
     "CreateDate": "2013-03-09T23:36:32Z",
     "Path": "/",
     "UserName": "james"
    }
  ]
}
```

Users[0]

JMESPathクエリーの基本的な表記方法

```
--query Users[0].[UserName,Path,UserId]
  {
   "Users": [
       {
        "Arn": "arn:aws:iam::XXXX:user/james",
        "UserId": "userid",
        "CreateDate": "2013-03-09T23:36:32Z",
        "Path": "/",
        "UserName": "james"
       }
    ]
  }
```

マルチセレクト

Users[0]

JMESPathクエリーの基本的な表記方法

```
--query Users[0].[UserName,Path,UserId]
```

```
[
  "james",
  "/",
  "userid"
]
```

JMESPathファンクション

クエリー種別	インプット	クエリー	結果
length	{ "people": [{"name": "a", }, {"name": "b", }, {"name": "c", }]}	length(people)	3
max_by	{"people": [{"name": "a", "age": 30}, {"name": "b", "age": 50}, {"name": "c", "age": 40}]}	max_by(people, &age).name	b
contains	{"myarray": ["foo", "bar", "foobar", "barfoobaz"]}	myarray[?contains(@, 'foo') == `true`]	["foo", "foobar", "barfoobaz"]

[options] : --query

フィルタリング前のJSON形式の出力

```
$ aws ec2 describe-instances
{
   "Reservations": [
      {
         "OwnerId": ”XXXXXXXXXXXX”,
         "ReservationId": "r-XXXXXXXX”,
         "Groups": [],
         "Instances": [
            {
               "Monitoring": {
                  "State": "enabled”
               },
…
```

[options] : --query

出力される項目（列）をフィルタリング

```
$ aws ec2 describe-instances --query 'Reservations[].Instances[].[InstanceId, InstanceType]'
[
    [
        "i-XXXXXXXX",
        "c3.2xlarge"
    ],
    [
        "i-XXXXXXXX",
        "t2.medium"
    ]
]
```

階層を絞込み

項目を絞込み

[options] : --query

出力される項目（列）をフィルタリングして、項目名を割り当て

```
$ aws ec2 describe-instances –query ¥
‘Reservations[].Instances[].{id:InstanceId, type:InstanceType}’
[
    {
        "type":"c3.2xlarge",
        "id":"i-XXXXXXXX"
    },
    {
        "type":"t2.micro",
        "id":"i-XXXXXXXX"
    },
...
```

項目名を追加

[options] : --query

出力されるアイテム（行）を完全一致でフィルタリング

```
$ aws ec2 describe-instances --query ¥
‘Reservations[].Instances[?InstanceType==`t2.micro`].[InstanceId, InstanceType][]’
[
    [
        "i-XXXXXXXX",
        "t2.micro"
    ]
]
```

アイテムを絞込み

[options] : --query

出力されるアイテム（行）を部分一致でフィルタリング

```
$ aws ec2 describe-instances --query ¥
‘Reservations[].Instances[?contains(InstanceType,`t2`)!=`true`].[InstanceId,
InstanceType][]’
[
    [
        "i-XXXXXXXX",
        ”t1.micro",
    ],
    [
        "i-XXXXXXXX”,
...
```

アイテムを絞込み

[options] : --query

出力されるアイテム（行）をAND条件でフィルタリング

```
$ aws ec2 describe-instances ¥
--query 'Reservations[].Instances[?State.Name!=`stopped`][] | ¥
[?InstanceType==`t2.micro`].[InstanceId, InstanceType, State.Name]'

[
    [
        "i-XXXXXXXX",
        "t2.micro",
        "running"
    ],
    [
        "i-XXXXXXXX”,
…
```

結果をパイプしてさらに
評価することでAND条件に

[options]： --query

特定タイプのインスタンス数を合計

```
$ aws ec2 describe-instances --query ¥
'length(Reservations[].Instances[?InstanceType==`t2.micro`][InstanceId][])‘
8
```

フィルターした配列数の合計を表示

[options]： --cli-input-json

2014年11月

CLIのパラメーターにJSONドキュメントを指定します。
ドキュメントは、String形式またはファイルで指定できます。

```
aws --cli-input-json (json) [options] <command> <subcommand>
```

JSONドキュメント

JSONドキュメントをファイル指定する場合、パラメーターに
file://プリフィックスを追加します。

```
$ aws ec2 run-instances --cli-input-json file://ec2runinst.json
```

カレントディレクトリのec2runinst.jsonを指定

http://docs.aws.amazon.com/cli/latest/userguide/generate-cli-skeleton.html

[options]： --generate-cli-skeleton

2014年11月

CLIパラメーターのJSONテンプレートを取得できます。

```
aws --generate-cli-skeleton [options] <command> <subcommand>
```

```
$ aws ec2 run-instances --generate-cli-skeleton
{
  "DryRun": true,
  "ImageId": "",
  "MinCount": 0,
  "MaxCount": 0,
  "KeyName": "",
  "SecurityGroups": [
      ""
  ],
...
```

http://docs.aws.amazon.com/cli/latest/userguide/generate-cli-skeleton.html

[subcommand]: Wait

2014年11月

AWS リソースの状態を待ち受けるためのサブコマンドです。先行コマンド実行後、AWSリソースの状態変更を待って後続処理をおこないたい場合に利用します。

```
aws  <command> wait <waiter name> [parameters]
```

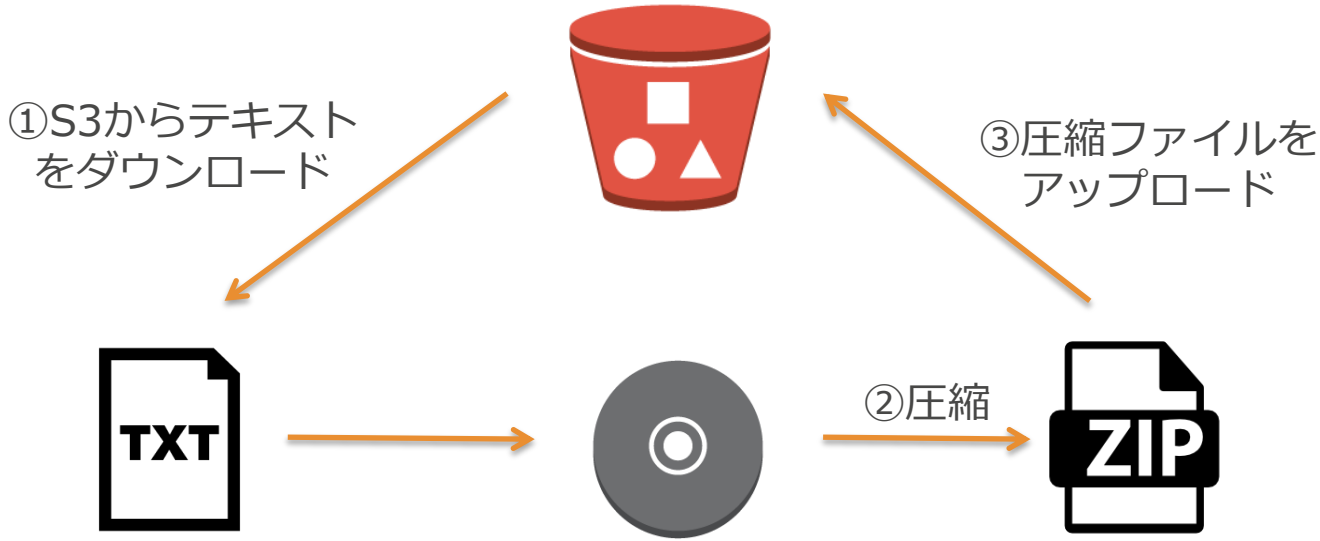
待ち受ける状態名

スナップショットの作成完了を待ってタグ付け

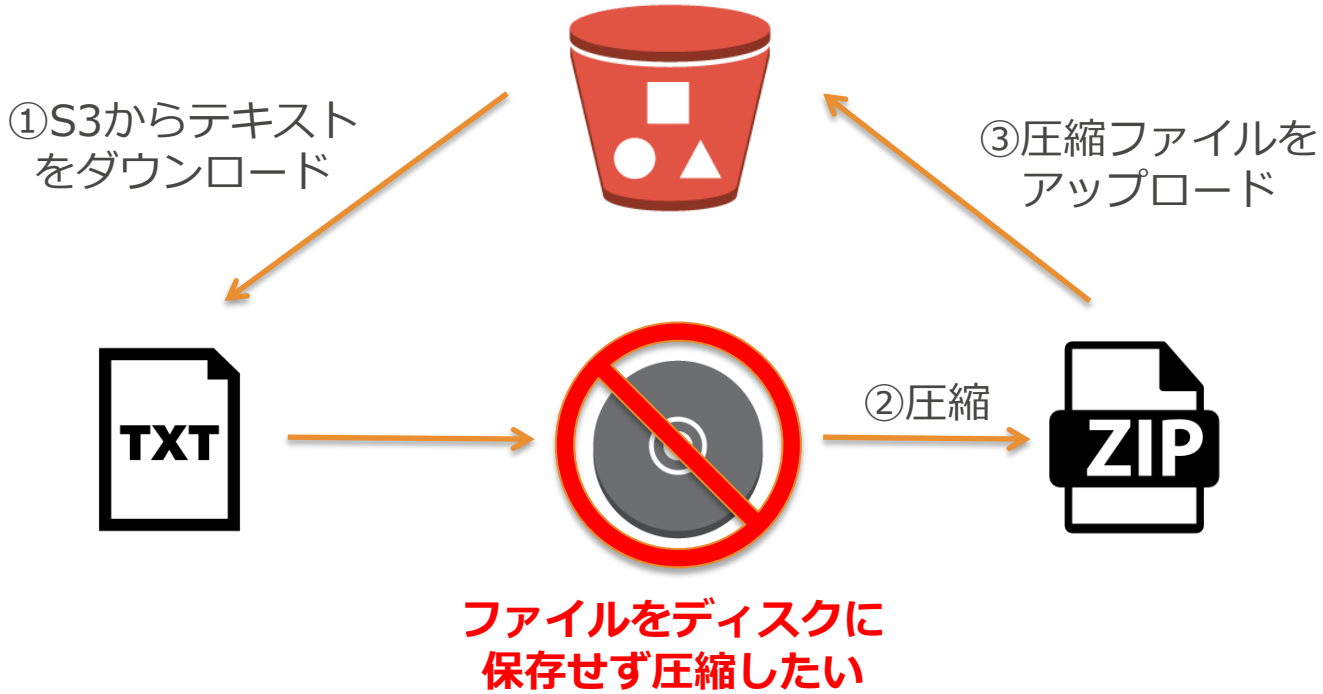
```
snapshot_id=$(aws ec2 create-snapshot --volume-id vol-XXXXXX --description "XXXXX" ¥
        --query SnapshotId --output text)

aws ec2 wait snapshot-completed --snapshot-ids $snapshot_id ; aws ec2 create-tags --
resources $snapshot_id --tags Key=Name,Value=SnapshotXXXX
```

http://docs.aws.amazon.com/cli/latest/reference/ec2/wait/index.html#cli-aws-ec2-wait

S3 Streaming



S3 Streaming



S3 Streaming

s3 cpコマンドでは、入出力パラメーターにハイフン"-"を指定することで、標準入出力とS3との間でダウンロード、アップロードが可能です。

```
aws s3 cp s3://bucket/key -
```

S3から標準出力へのダウンロード

```
aws s3 cp - s3://bucket/key
```

標準入力からS3へのアップロード

ディスクを介さずS3のファイルを圧縮

```
$ aws s3 cp s3://bucket/key - | bzip2 -best | aws s3 cp - s3://bucket/key.bz2
```

S3->標準出力　　標準出力->zip化　　標準入力->S3

http://docs.aws.amazon.com/cli/latest/reference/s3/cp.html

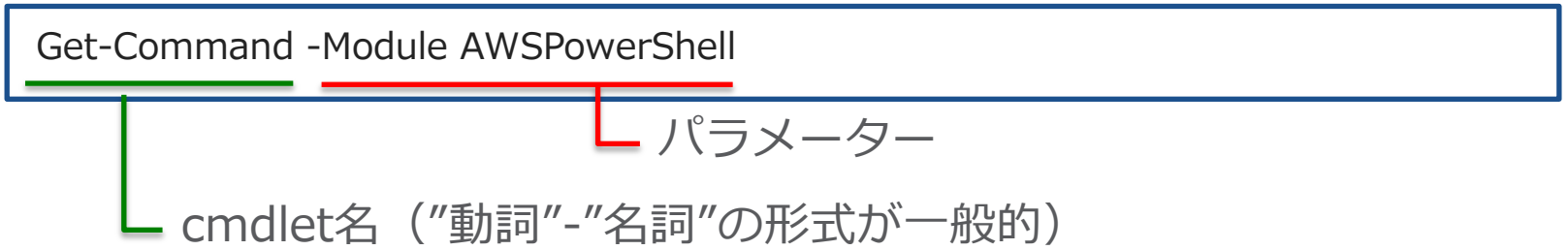
AWS Tools for Windows PowerShell

Windows PowerShellの基本

Microsoft社が開発したCLIシェル／スクリプト言語

- コマンドレット(cmdlet)と呼ばれる単位でプログラムを管理
- オブジェクト指向言語
- .NET Frameworkクラス・ライブラリを利用可能

文法



コマンドレットの検索

Get-Commandコマンドレットは、インストールされたモジュール内のコマンドレットを検索することができます。

"AWSPowerShell"モジュールのコマンドレットの中から、コマンドレット名に”EC2”という文字列が含まれるものを取得

```
PS C:¥> Get-Command -Module AWSPowerShell -Name “*EC2*”
CommandType     Name                                               Definition
-----------     ----                                               ----------
Cmdlet          Add-EC2ClassicLinkVpc                                 ...
Cmdlet          Add-EC2InternetGateway                                ...
Cmdlet          Add-EC2NetworkInterface                               ...
```

コマンドレットのHelp

Get-Helpコマンドレットは、コマンドレットのHelpを表示できます。

"Get-Command"コマンドレットのHelp

```
PS C:¥Users¥yukichib> Get-Help Get-Command

NAME
    Get-Command

SYNOPSIS
    コマンドレットおよびその他の Windows PowerShell コマンド要素に関する基本情報を取得します。

SYNTAX
    Get-Command [[-Name] <string[]>] [-CommandType {Alias | Function | Filter | Cmdlet |
ExternalScript | Application | Script | All}] [[-ArgumentList] <Object[]>] [-Module <string[]>] [-
Syntax] [-TotalCount <int>] [<CommonParameters>]
```

出力オブジェクト

コマンドレット実行結果は、.NET Frameworkオブジェクトの配列として出力されます。

```
PS C:¥> Get-EC2AvailabilityZone | ft *

Region          ZoneState Message  Messages RegionName     State     ZoneName
                                                                     プロパティ名
------          --------- -------  -------- ----------     -----     --------
ap-northeast-1 available  {}       {}       ap-northeast-1 available ap-northeast-1a
ap-northeast-1 available  {}       {}       ap-northeast-1 available ap-northeast-1c
                                                Amazon.EC2.Model.AvailabilityZoneクラスの
                                                2 つのオブジェクトの配列として出力
```

パイプライン

実行結果をパイプライン”|”で次のコマンドレットに渡すことができます。

"AWSPowerShell"モジュールのコマンドレットを全て取得し、その中から
コマンドレット名が“Get-EC2Instance”のものを絞り込んで表示

```
PS C:¥> Get-Command -Module AWSPowerShell | `
>> Where-Object {$_.Name -eq "Get-EC2Instance"}
CommandType     Name                                               Definition
-----------     ----                                               ----------
Cmdlet          Get-EC2Instance                                    ...
```

※Powerhshellではスクリプト中の改行にはバッククォート"`"を使います。

オブジェクト構造の確認

Get-Memberコマンドレットは、クラスで定義されたプロパティとメソッドを表示します。

EC2インスタンスオブジェクトのプロパティを表示

```
PS C:¥> (Get-EC2Instance).Instances | Get-Member -MemberType Property | Select Name,Definition

Name                 Definition
----                 ----------
AmiLaunchIndex       int AmiLaunchIndex {get;set;}
Architecture         Amazon.EC2.ArchitectureValues Architecture {get;set;}
ClientToken          string ClientToken {get;set;}
EbsOptimized         bool EbsOptimized {get;set;}
Hypervisor           Amazon.EC2.HypervisorType Hypervisor {get;set;}
ImageId              string ImageId {get;set;}
InstanceId           string InstanceId {get;set;}
...
```

フォーマット指定

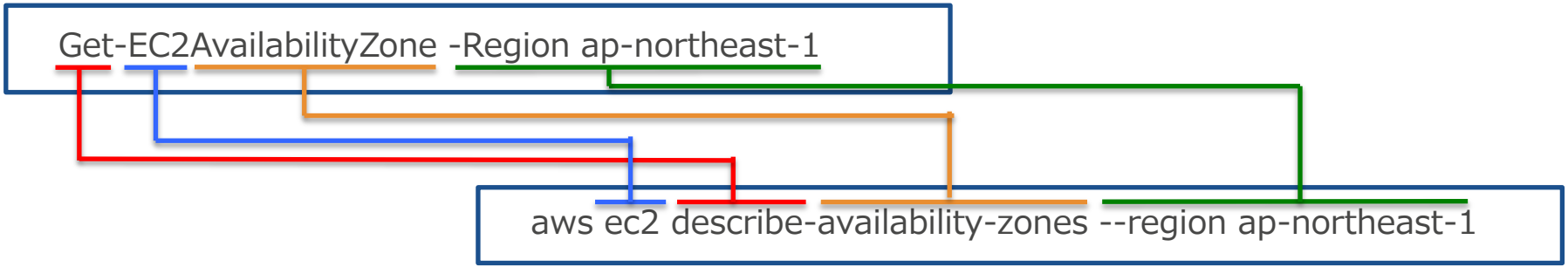
Format-List(fl)、Format-Table(ft)などのコマンドレットを利用して、表示形式を指定することができます。

```
PS C:¥> $EC2Instances =  (Get-EC2Instance).Instances |select InstanceType, LaunchTime

PS C:¥> $EC2Instances | fl

InstanceType : m3.large
LaunchTime   : 2015/07/09 16:34:01

InstanceType : t2.micro
LaunchTime   : 2015/04/08 16:47:30
....
```

```
PS C:¥> $EC2Instances | ft

InstanceType     LaunchTime
------------     ----------
m3.large         2015/07/09 16:34:01
t2.micro         2015/04/08 16:47:30
....
```

AWS PowerShell コマンドレット

基本的な使い方はAWS CLIと同様です。
コマンドレット名は [動詞]-[サービス名][名詞]になっています。



代表的なアクション動詞のAWS APIとの対応は以下のとおりです。

	読み取り	新規作成	上書き	削除
API	describe, list	create	put	delete
PowerShell	Get	New	Write	Remove

Cmdlet Reference: http://docs.aws.amazon.com/powershell/latest/reference/Index.html

Get-AWSCmdletName

2015年2月

AWS APIやAWS CLIとの対応を確認するコマンドレットです。

APIアクションからコマンドレット名を取得

```
PC C:¥> Get-AWSCmdletName –ApiOperation DescribeInstances -Service EC2
CmdletName          ServiceOperation     ServiceName                    CmdletNounPrefix
----------          ----------------     -----------                    ----------------
Get-EC2Instance     DescribeInstances    Amazon Elastic Compute Cloud   EC2
```

CLIコマンドからコマンドレット名を取得

```
PC C:¥> Get-AWSCmdletName -AwsCliCommand "aws ec2 describe-instances"
CmdletName          ServiceOperation     ServiceName                    CmdletNounPrefix
----------          ----------------     -----------                    ----------------
Get-EC2Instance     DescribeInstances    Amazon Elastic Compute Cloud   EC2
```

http://docs.aws.amazon.com/ja_jp/powershell/latest/userguide/pstools-discovery-aliases.html

パイプラインの使い方

パイプラインを利用して、AWSリソースの取得／操作をワンラインで操作できます。

取得したEC2インスタンスを全て停止

```
PC C:¥> Get-EC2Instance | Stop-EC2Instance
```

EC2インスタンスオブジェクトの配列をStop-EC2Instanceの第一引数(-Instance)に渡して実行

取得したAWSリージョン内の全AMIを取得

```
PC C:¥>  Get-AWSRegion | % { Get-EC2Image -Owner self -Region $_ }
```

リージョンオブジェクトの配列をGet-EC2Imageの-Region引数に渡して実行

※PowerShellコマンドレットの引数にはポジションが設定されており、明示的に引数を指定しない場合、先行処理の結果がパイプラインの後続コマンドレットの第一引数の値として評価されます。

http://docs.aws.amazon.com/ja_jp/powershell/latest/userguide/pstools-discovery-aliases.html

[options] : -Filter

AWSPowershellコマンドレットの出力を、Filterクラスのオブジェクトを用いてフィルタリングする仕組みです。

Nameタグの値が”TEST”のEC2インスタンスを取得

```
PC C:¥>  $tag = New-Object Amazon.EC2.Model.Filter -Property @{Name="tag:Name";Values="TEST"}
```

フィルターオブジェクトを作成

```
PC C:¥>  Get-EC2Instance -Filter $tag
```

-Filterでフィルターオブジェクトを指定

グループ名が”SG1”,"SG2"のセキュリティグループを取得

```
PC C:¥>  $SGNames = New-Object Amazon.EC2.Model.Filter -Property `
>> @{Name="group-name";Values=@("SG1","SG2")}
PC C:¥>  Get-EC2SecurityGroup -Filter $SGNames
```

Get-AWSPublicIpAddressRange

2015年1月

AWSが利用するパブリックIPアドレスレンジを取得するコマンドレットです。AWS CLIなどでは、ip-ranges.jsonをダウンロードする必要がありますが、AWS PowerShellではコマンドレットが用意されています。

```
PS C:¥> Get-AWSPublicIpAddressRange
IpPrefix                Region      Service
--------                ------      -------
50.19.0.0/16            us-east-1   AMAZON
...
50.19.0.0/16            us-east-1   EC2
...
205.251.192.0/21        GLOBAL      ROUTE53
54.232.40.64/26         sa-east-1   ROUTE53_HEALTHCHECKS
...
204.246.176.0/20        GLOBAL      CLOUDFRONT
...
```

http://docs.aws.amazon.com/powershell/latest/reference/Index.html
http://docs.aws.amazon.com/ja_jp/general/latest/gr/aws-ip-ranges.html

ConvertFrom-Json

ConvertFrom-Jsonは、JSONドキュメントを.NETオブジェクトに変換するPowerShell標準コマンドレットです。CloudTrailログなどのJSON形式のドキュメントを扱うのに便利なコマンドレットです。

S3バケットからCloudTrailログを取得し.NETオブジェクトで出力

```
PS C:¥> $Bucket = "CloudTrailLogBucketName"; $Key = "CloudTrailLogFileName"
PS C:¥> $URL=Get-S3PreSignedURL -Key $Key -BucketName $Bucket -Expires (Get-Date).AddMinutes(1)
```

署名付きURLを発行

```
PS C:¥> $LogFiles = (wget $URL).content|ConvertFrom-Json
```

ログを取り出しオブジェクトに変換

```
PS C:¥> $LogFiles.records | select eventTime,EventName
```

オブジェクトをフィルタリングして出力

```
eventTime            eventName
---------            ---------
2015-07-15T00:13:14Z DescribeLoadBalancers
2015-07-15T00:46:51Z ListRolePolicies
...
```

Export-CSV

Export-CSVコマンドレットは、.NETオブジェクトをCSV形式に変換してファイル出力するPowerShell標準コマンドレットです。AWSリソース一覧を取得する際に便利なコマンドレットです。

EC2のNameタグ、インスタンスID、インスタンスタイプ、AZ一覧をCSVで取得

```
PS C:¥> $EC2=(Get-EC2Instance).Instances | `
>> select @{Name="NameTag"; Expression={($_.tag | where {$_.key -eq "Name"}).value}}, `
>> instanceId,InstanceType,@{Name="AZ"; Expression={($_.Placement).AvailabilityZone}},VpcId

PS C:¥> $EC2 | Export-CSV C:¥ec2list.csv
```

項目をフィルタリング

CSV形式でファイル出力

まとめ

- AWSコマンドラインインターフェース（CLI）
  - AWSサービスを管理するための統合ツール
  - Windows/Mac/Linuxなどマルチプラットフォームに対応
  - Queryオプションを利用して様々な形式でコマンドの出力を操作可能

- AWS Tools for Windows PowerShell
  - PowerShellからAWSのサービスを管理するためのモジュール
  - Get-AWSCmdletNameコマンドレットで対応するAPI,CLIを確認可能
  - PowerShell標準コマンドレットを活用してAWSの管理を効率化可能

JAWS-UG CLI専門支部

～CLIに取り組む仲間が参加しています！～

Facebookグループ https://www.facebook.com/groups/jaswug.cli/
イベント告知ページ https://jawsug-cli.doorkeeper.jp/

Q&A

次回Webinarのお申し込み

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

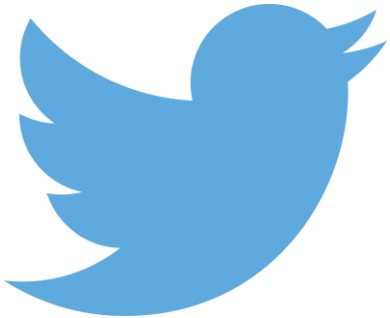
もしくは
http://on.fb.me/1vR8yWm

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！

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

次回のAWS Black Belt Tech Webinar は、

7月29日 18:00～

AWS CloudHSM & AWS KMS

ご参加ありがとうございました。