Vertical Scaling for the AWS Ops Automator

AWS Implementation Guide

February 2019

Last updated: March 2019 (see revisions)
Contents
Overview ........................................................................................................................................... 3
Architecture Overview ...................................................................................................................... 4
Components ..................................................................................................................................... 5
Scaling Method ................................................................................................................................ 5
  Resizing an Existing Instance ........................................................................................................ 5
  Replacing with a New Instance of a Different Size ....................................................................... 5
Resized Instance Type Range .......................................................................................................... 6
Assumed Instance Type ..................................................................................................................... 6
Instance Tagging ............................................................................................................................... 6
Considerations ................................................................................................................................... 7
  Cross-Account and Cross-Region Scaling .................................................................................... 7
Amazon EC2 Limits ............................................................................................................................ 7
Automated Deployment .................................................................................................................... 7
  What We’ll Cover ............................................................................................................................ 8
  Step 1. Launch the Ops Automator Stack in the Primary Account ............................................ 8
  Step 2. Launch the Event Forwarder Template in Secondary Account(s) (Optional) ............... 10
  Step 3. Launch the Role Template in Secondary Account(s) (Optional) ................................. 11
  Step 4. Launch the Vertical Scaling Template in the Primary Account .................................... 12
Appendix A: Scaling Limitations ....................................................................................................... 14
Appendix B: Troubleshooting ........................................................................................................... 14
Appendix C: Tag and Parameter Placeholders .............................................................................. 14
Appendix D: Collection of Operational Metrics .............................................................................. 16
Document Revisions ......................................................................................................................... 18

About This Guide
This implementation guide discusses steps for enabling vertical scaling for the Amazon Web Services (AWS) Ops Automator solution. It includes links to AWS CloudFormation templates
that launch, configure, and run the AWS services required to deploy vertical scaling, using AWS best practices for security and availability.

The guide is intended for IT infrastructure architects, administrators, and DevOps professionals who have practical experience architecting on the AWS Cloud.

Overview

The AWS Ops Automator helps you more easily configure automated tasks on your AWS resources. The solution now features vertical scaling for Amazon Elastic Compute Cloud (Amazon EC2) instances. With vertical scaling, the solution automatically adjusts capacity to maintain steady, predictable performance at the lowest possible cost. The solution can scale your instances by resizing your existing instance, or by replacing your existing instances with a new instance of a different size. With this update, the AWS Ops Automator can help make setting up vertical scaling easier.

**Note:** To take advantage of the vertical scaling functionality, you must deploy the latest version of the Ops Automator as a new stack. You cannot update a previously deployed version of the solution stack.
Architecture Overview

Deploying vertical scaling for the AWS Ops Automator builds the following environment in the AWS Cloud.

To set up vertical scaling, you must deploy the AWS Ops Automator solution, which creates a core framework for automated tasks, in the primary account. The solution includes an AWS CloudFormation template that sets up the necessary tasks for vertical scaling. For more information about the Ops Automator, see the implementation guide.

For vertical scaling, the Ops Automator event handler AWS Lambda function retrieves the configuration for the CPU metrics task from Amazon DynamoDB and executes that task at a customer-defined interval. The CPU metrics task checks the CPU utilization metrics of your Amazon Elastic Compute Cloud (Amazon EC2) instances against thresholds you define during deployment. When CPU utilization is below the low threshold, the instance is tagged to be scaled down. When utilization is above the high threshold, the instance is tagged to be scaled up. If utilization is within both thresholds, the instance is not changed.
When an instance is tagged, an Amazon CloudWatch event triggers the *vertical scaling* task to resize the existing instance, or replace the instance with a new instance of a different size, depending on which configuration you choose. You can specify a range of instance types for resizing.

The solution also includes AWS CloudFormation templates you can deploy in secondary accounts and AWS Regions to set up cross-account and cross-region vertical scaling. For more information, see [Cross-Account and Cross-Region Scaling](#).

**Components**

**Scaling Method**

This solution includes two methods for scaling, resizing your existing instance or replacing your instance with a new instance of a different size. When you specify a vertical scaling method, the solution creates the applicable task (*resize* or *replace*).

**Resizing an Existing Instance**

When you choose to resize your existing instances, the solution automatically stops your existing instance, resizes the instance to the next defined size up or the next defined size down, then starts the instance again.

During the resizing, Amazon Elastic Block Store (Amazon EBS) volumes on the instance will remain attached and the data will persist. However, any data on the ephemeral storage of your instance will be lost. To keep your data, it must be stored on an attached Amazon EBS volume.

This approach may be best for applications that can experience downtime, and use cases where the data on volumes must be preserved.

**Replacing with a New Instance of a Different Size**

When you choose to replace your existing instance with a new instance of a different size, the solution automatically launches a new instance with the same Amazon Machine Image and settings that is the next defined size up or the next defined size down. The solution is integrated with Elastic Load Balancing to automatically register the new instance with the same load balancer or target group.

Amazon EBS volumes on the original instance will be deleted, so any data on those volumes will be lost.
This approach may be best for applications that cannot experience downtime, and use cases where the data on volumes does not need to be preserved.

**Resized Instance Type Range**

This solution enables you to specify a range of Amazon Elastic Compute Cloud (Amazon EC2) instance types for scaling. You must specify at least two valid Amazon EC2 instance types, and the instances types must be compatible. For more information, see Compatibility for Resizing Instances in the Amazon EC2 User Guide for Linux Instances.

The solution will not scale to instances that are not within the defined range. If an instance that exceeds the high threshold is at the top of the instance type range, the solution will not scale the instance and a message will be logged in Amazon CloudWatch. If an instance that falls below the low threshold is at the bottom of the range, the solution will not scale the instance and a message will be logged in CloudWatch.

**Assumed Instance Type**

You can use the Assumed Type AWS CloudFormation parameter to specify an instance type to use for instances that are not within the range you define. This instance type will be used as the basis for scaling.

For example, a customer might specify a range of t2.micro, t2.small, and t2.medium, and an assumed type of t2.small. If that customer has a t3.small instance that exceeds the high threshold, the solution will use t2.small as the original instance type and scale the instance to t2.medium.

If you do not specify an assumed type, the solution will log an error in Amazon CloudWatch when it has to scale an instance that is not within the range you defined.

**Instance Tagging**

After an Amazon EC2 instance is scaled, the solution tags the instance with a description of the most recent scaling action. The tag provides visibility and associates resizing actions with the specific resources that were affected.

Instances are tagged with the following tag: LastScalingAction=Instance scaled from type {org-instance-type} to type {new-instance-type} at {iso-datetime} by task {task} in Ops Automator stack {stack}.

The following table includes the placeholders and their definitions.
Considerations

Cross-Account and Cross-Region Scaling
Vertical scaling for the AWS Ops Automator enables you to scale Amazon Elastic Compute Cloud (Amazon EC2) instances across accounts and AWS Regions. To use this functionality, you must deploy the event forwarder AWS CloudFormation template in each applicable account and region. You must also deploy the account role configuration template in each secondary account. We recommend using AWS CloudFormation StackSets to deploy these stacks in an easy and consistent way.

The event forwarder forwards scaling events across accounts and regions. The account role configuration creates a trust relationship with the AWS Identity and Access Management role in the AWS Ops Automator stack that allows the solution’s Lambda function to make the required API calls across accounts.

Amazon EC2 Limits
Each AWS account has Amazon EC2 service limits. These limits apply when this solution restarts an instance with a new size or replaces an instance with a new, resized instance. When a limit is reached, the solution will not scale instances and will log an error in CloudWatch.

Automated Deployment
Before you launch the solution, please review the information discussed in this guide and the AWS Ops Automator implementation guide. Follow the step-by-step instructions in this section to configure and deploy the solution.

Time to deploy: Approximately 15 minutes
What We’ll Cover

The procedure for deploying this architecture on AWS consists of the following steps. For detailed instructions, follow the links for each step.

**Step 1. Launch the AWS Ops Automator Stack**
- Launch the AWS CloudFormation template into your primary AWS account.
- Enter values for required parameters: **Stack Name**.
- Review the other template parameters, and adjust if necessary.

**Step 2. Launch the Event Forwarder Template in Secondary Account(s) (Optional)**
- Launch the event forwarder AWS CloudFormation template into the secondary account with applicable resources.
- Enter values for required parameters: **Stack Name**.
- Review the template parameters, and adjust if necessary.

**Step 3. Launch the Role Template in Secondary Account(s) (Optional)**
- Launch the role AWS CloudFormation template into the secondary account with applicable resources.
- Enter values for required parameters: **Stack Name**.
- Review the template parameters, and adjust if necessary.

**Step 4. Launch the Vertical Scaling Template in the Primary Account**
- Launch the vertical scaling task AWS CloudFormation template into the primary account.
- Review the template parameters, and adjust if necessary.

**Step 1. Launch the Ops Automator Stack in the Primary Account**

**Note:** You are responsible for the cost of the AWS services used while running this solution. For full details, see the pricing webpage for each AWS service you will be using in this solution.

1. Log in to the AWS Management Console and click the button to the right to launch the **ops-automator** AWS CloudFormation template.
   You can also download the template as a starting point for your own implementation.
2. The template is launched in the US East (N. Virginia) Region by default. To launch the solution in a different AWS Region, use the region selector in the console navigation bar.

3. On the Select Template page, verify you selected the correct template and choose Next.

4. On the Specify Details page, specify a name for your solution stack.

5. Under Parameters, review the parameters for the template and modify them as necessary. This solution uses the following default values.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ops Automator Tag Name</td>
<td>OpsAutomatorTaskList</td>
<td>The tag key (name) that identifies applicable resources. The tag value will contain the list of tasks to be performed on tagged resources.</td>
</tr>
<tr>
<td>Enable CloudWatch Metrics</td>
<td>Yes</td>
<td>Choose whether to collect Amazon CloudWatch metrics for the AWS Ops Automator. Detailed metrics for individual tasks can be configured at the task-level.</td>
</tr>
<tr>
<td>Schedule active?</td>
<td>Yes</td>
<td>Choose whether to activate the scheduling task feature</td>
</tr>
<tr>
<td>Clean up task tracking table?</td>
<td>Yes</td>
<td>Choose whether to clean the task tracking table</td>
</tr>
<tr>
<td>Export task tracking table?</td>
<td>No</td>
<td>Choose whether to export the task tracking table to Amazon S3</td>
</tr>
<tr>
<td>Keep failed tasks?</td>
<td>Yes</td>
<td>Choose whether to store failed tasks in the Amazon DynamoDB table</td>
</tr>
<tr>
<td>How long to keep tasks?</td>
<td>168</td>
<td>The number of hours to keep a task before it is automatically deleted</td>
</tr>
<tr>
<td>Days to keep configuration backups</td>
<td>7</td>
<td>The number of days to keep configuration backups before they are automatically deleted</td>
</tr>
<tr>
<td>Log Retention Days</td>
<td>30</td>
<td>The number of days to keep logs before they are automatically deleted</td>
</tr>
<tr>
<td>Send Anonymous Usage Data</td>
<td>Yes</td>
<td>Send anonymous data to AWS to help us understand AWS Ops Automator usage across our customer base as a whole. To opt out of this feature, select No. For more information, see Appendix D.</td>
</tr>
</tbody>
</table>

6. Choose Next.

7. On the Options page, choose Next.

8. On the Review page, review and confirm the settings. Select the checkbox acknowledging that the template will create AWS Identity and Access Management (IAM) resources.

9. Choose Create to deploy the stack.
You can view the status of the stack in the AWS CloudFormation Console in the **Status** column. You should see a status of **CREATE_COMPLETE** in approximately 10 minutes.

**Step 2. Launch the Event Forwarder Template in Secondary Account(s) (Optional)**

Use this procedure to scale instances across accounts and AWS Regions. Launch this template in each applicable account and each applicable region.

**Important:** To scale instances across accounts and regions, you must deploy the event forwarder AWS CloudFormation template (**AccountForwardEvents**) in each applicable account and region, and you must deploy the account role configuration template (**AccountRoleConfiguration**) in each account.

1. In the primary account’s Amazon S3 console, navigate to the bucket for the AWS Ops Automator solution stack.

   **Note:** You can find the name of the S3 bucket in the AWS CloudFormation stack outputs tab. The bucket name is value of the **ConfigurationBucketName** key.

2. In the **Accounts Configuration** folder, select the **AccountForwardEvents** AWS CloudFormation template.

3. Copy the **Link** value.

4. In the **AWS CloudFormation StackSets console**, select **Create StackSet**.

5. Select **Specify an Amazon S3 template URL**.

6. Paste the template link into the text box and select **Next**.

7. Enter a **StackSet name**.

8. Under **Parameters**, verify that the **EC2 Tag Change Events** parameter is set to **Yes**.

   **Note:** For vertical scaling, the **EC2 Tag Change Events** parameter must be set to **Yes**. The other parameters do not affect vertical scaling.

9. Select **Next**.

10. Select **Next**. Then, on the **Review** page, review and confirm the settings. Select the checkbox acknowledging that the template will create AWS Identity and Access Management (IAM) resources.

11. Choose **Create** to deploy the stack.
Step 3. Launch the Role Template in Secondary Account(s) (Optional)

Use this procedure to configure the cross-account role required to scale instances across accounts and regions. You must complete step 2 before you complete this step. Launch this template in each secondary account.

**Important:** To scale instances across accounts and regions, you must deploy the event forwarder AWS CloudFormation template (AccountForwardEvents) in each applicable account and region, and you must deploy the account role configuration template (AccountRoleConfiguration) in each account.

1. In the **Accounts Configuration** folder, select the AccountRoleConfiguration AWS CloudFormation template.
2. Copy the **Link** value.
3. In the AWS CloudFormation StackSets console, select **Create StackSet**.
4. Select **Specify an Amazon S3 template URL**.
5. Paste the template link into the text box and select **Next**.
6. Enter a **StackSet name**.
7. Under **Parameters**, verify that the **EC2 Tag Instance By CPU Utilization** parameter is set to **Yes**.
8. Select the scaling method.
   - If you selected **resize** as your **Scaling Method** when you launched the Resize Task Template, set **EC2Resize Instance** to **Yes**.
   - If you selected **replace** as your **Scaling Method** when you launched the Resize Task Template, set **EC2 Replace Instance** to **Yes**.

**Note:** For vertical scaling, the **EC2 Tag Instance By CPU Utilization** parameter and the applicable scaling method parameter must be set to **Yes**. The other parameters do not affect vertical scaling.

9. Verify that there is no role specified in the **Custom Role** parameter.
10. Select **Next**.
11. Select **Next**. Then, on the **Review** page, review and confirm the settings. Select the checkbox acknowledging that the template will create AWS Identity and Access Management (IAM) resources.
12. Choose **Create** to deploy the stack.

**Step 4. Launch the Vertical Scaling Template in the Primary Account**

1. In the primary account’s Amazon S3 console, navigate to the bucket for the AWS Ops Automator solution stack.

   **Note:** You can find the name of the S3 bucket in the AWS CloudFormation stack outputs tab. The bucket name is value of the `ConfigurationBucketName` key.

2. In the **Task Configuration/Scenario Templates** folder, select the `Ec2VerticalScaling` AWS CloudFormation template.

3. Copy the **Link** value.

4. In the AWS CloudFormation console, select **Create Stack**.

5. Select **Specify an Amazon S3 template URL**.

6. Paste the template link into the text box and select **Next**.

7. Enter a **Stack name**.

   **Note:** This name will also be the name of the scaling plan that is used to tag Amazon EC2 instances for scaling.

8. Under **Parameters**, review the parameters for the template and modify them as necessary.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ops Automator Stack</td>
<td><code>&lt;stackname&gt;</code></td>
<td>The name of the Ops Automator AWS CloudFormation stack. This parameter is auto-populated with the name you chose for your Ops Automator stack.</td>
</tr>
<tr>
<td>Scaling Interval</td>
<td>5</td>
<td>Specify how often, in minutes, to check CPU utilization metrics.</td>
</tr>
<tr>
<td>Scaling Method</td>
<td><code>&lt;Requires input&gt;</code></td>
<td>Choose whether to restart your instances with a new size, or replace your instance with a new, resized instance. Select <em>resize</em> or <em>replace</em>.</td>
</tr>
<tr>
<td>Instance Types</td>
<td>t2.micro, t2.small, t2.medium, t2.large</td>
<td>A comma-delimited list of valid, compatible instance types. Note: The list must be sorted from smallest to largest and must contain at least two instance types.</td>
</tr>
<tr>
<td>Assumed Type</td>
<td>t2.small</td>
<td>The default instance type that will be used if the instance type of an existing instance is not within the range defined in the <strong>Instance Types</strong> parameter.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Default</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>---------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>CPU High Utilization Threshold</td>
<td>90</td>
<td>The high threshold for average CPU utilization that will determine whether an instance is scaled up</td>
</tr>
<tr>
<td>CPU Low Utilization Threshold</td>
<td>10</td>
<td>The low threshold for average CPU utilization that will determine whether an instance is scaled down</td>
</tr>
<tr>
<td>Instance tags</td>
<td></td>
<td>Tags to set on the resized Amazon EC2 instance</td>
</tr>
<tr>
<td>Account List</td>
<td>&lt;Optional input&gt;</td>
<td>A comma-delimited list of account IDs for instance resizing</td>
</tr>
<tr>
<td>Region List</td>
<td>&lt;Optional input&gt;</td>
<td>A comma-delimited list of AWS Regions for instance resizing</td>
</tr>
</tbody>
</table>

**Note:** You must enable the `Ec2TagCpuInstance`, `Ec2ReplaceInstance`, and `Ec2ResizeInstance` actions in the cross-account role in each account. For more information, see Cross-Account and Cross-Region Scaling.

**Note:** You must launch the event forwarder stack in each applicable region. For more information, see Cross-Account and Cross-Region Scaling.

**Note:** You must specify an instance type that is specified in the Instance Types parameter.

Detailed Logging

Choose whether to generate detailed Amazon CloudWatch logs

9. Select **Next**.

10. Select **Next**. Then, on the **Review** page, review and confirm the settings. Select the checkbox acknowledging that the template will create AWS Identity and Access Management (IAM) resources.

11. Choose **Create** to deploy the stack.
Appendix A: Scaling Limitations

Vertical scaling for the AWS Ops Automator logs error messages in Amazon CloudWatch if the solution is unable to scale an Amazon Elastic Compute Cloud (Amazon EC2) instance. The solution will not scale instances in the following scenarios:

- **When scaling up or down requires an instance type outside of the specified range.** For more information, see [Resized Instance Type Range](#).

- **When the maximum number of on-demand instances allowed per account is reached.** For more information on limits, see [Amazon EC2 Service Limits](#) in the [Amazon EC2 User Guide for Linux Instances](#).

- **When the original instance type is not compatible with the new instance type.** For more information, see [Compatibility for Resizing Instances](#) in the [Amazon EC2 User Guide for Linux Instances](#).

Appendix B: Troubleshooting

If there are permission errors in the solution’s log streams, cross-account and cross-region scaling are not set up properly. Verify that the Ec2TagCpuInstance, Ec2ReplaceInstance, and Ec2ResizeInstance are allowed.

If instances in secondary accounts are being tagged but not scaled, verify that the event forwarder stack has been deployed for the secondary account and/or region.

Appendix C: Tag and Parameter Placeholders

Placeholders can be used in parameters such as names, prefixes, and descriptions, and tag names and values set by AWS Ops Automator actions. Placeholders have the format `{name}`. When tasks are executed, placeholders are replaced with dynamic values. This allows the solution to give created resources dynamic names and descriptions, and to set dynamic tag names and values on created or affected resources.

The following table contains a list of common placeholders.

<table>
<thead>
<tr>
<th>Placeholder</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>{account}</td>
<td>The account in which the task is executed</td>
</tr>
<tr>
<td>{stack}</td>
<td>The name of the Ops Automator stack</td>
</tr>
<tr>
<td>{date}</td>
<td>The date in YYYYMMDD format</td>
</tr>
<tr>
<td><strong>Placeholder</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>----------------------</td>
<td>--------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>{datetime}</td>
<td>The date in YYYYMMDDHHMMSS format</td>
</tr>
<tr>
<td>{day}</td>
<td>The day in format DD</td>
</tr>
<tr>
<td>{hour}</td>
<td>The hour in format HH</td>
</tr>
<tr>
<td>{iso-date}</td>
<td>The date in ISO format</td>
</tr>
<tr>
<td>{iso-datetime}</td>
<td>The date and time in ISO format</td>
</tr>
<tr>
<td>{iso-time}</td>
<td>The time in ISO format</td>
</tr>
<tr>
<td>{iso-weekday}</td>
<td>The weekday in ISO format</td>
</tr>
<tr>
<td>{minute}</td>
<td>The minute in MM format</td>
</tr>
<tr>
<td>{month}</td>
<td>The month in MM format</td>
</tr>
<tr>
<td>{monthname}</td>
<td>The abbreviated name of the month. For example, Jan.</td>
</tr>
<tr>
<td>{monthname-long}</td>
<td>The full name of the month. For example, January.</td>
</tr>
<tr>
<td>{region}</td>
<td>The region in which the task is executed</td>
</tr>
<tr>
<td>{second}</td>
<td>The seconds in SS format</td>
</tr>
<tr>
<td>{task-tag}</td>
<td>Expands to the name of the tag configured for the task list. For example, OpsAutomatorTaskList</td>
</tr>
<tr>
<td>{task}</td>
<td>The name of the task that resized or replaced the instance</td>
</tr>
<tr>
<td>{task-id}</td>
<td>The unique ID of the executed task</td>
</tr>
<tr>
<td>{task-group}</td>
<td>The unique ID of the task group. A task group is a collection of tasks that are the result of a task started by the scheduler. One task can have multiple task executions depending on the aggregation level of the task. Each task will have its own task ID, but the tasks will share a group ID.</td>
</tr>
<tr>
<td>{time}</td>
<td>The time in HHMMSS format</td>
</tr>
<tr>
<td>{timezone}</td>
<td>The time zone configured for the task</td>
</tr>
<tr>
<td>{weekday}</td>
<td>The abbreviated name of the weekday. For example, Mon.</td>
</tr>
<tr>
<td>{weekday-long}</td>
<td>The full name of the weekday. For example, Monday.</td>
</tr>
<tr>
<td>{year}</td>
<td>The year in YYYY format</td>
</tr>
</tbody>
</table>

Ops Automator actions can have action-specific placeholders for tags.

The following table contains placeholders that are specific to the `Ec2ReplaceInstance` and `Ec2ResizeInstance` actions.
The following table contains a list of special placeholder.

<table>
<thead>
<tr>
<th>Placeholder</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>{instance-id}</td>
<td>The instance ID of the volume from which the snapshot was taken</td>
</tr>
<tr>
<td>{volume-id}</td>
<td>The volume ID of the volume from which the snapshot was taken</td>
</tr>
<tr>
<td>{device}</td>
<td>The device of the volume from which the snapshot was taken</td>
</tr>
<tr>
<td>{snapshot-ids}</td>
<td>The snapshot IDs for snapshots taken for all volumes of an instance</td>
</tr>
<tr>
<td>{snapshot-id}</td>
<td>The snapshot ID for a snapshot taken from a volume</td>
</tr>
</tbody>
</table>

Appendix D: Collection of Operational Metrics

This solution includes an option to send anonymous operational metrics to AWS. We use this data to better understand how customers use this solution and related services and products. When enabled, the following information is collected and sent to AWS:

- **Solution ID**: The AWS solution identifier
- **Unique ID (UUID)**: Randomly generated, unique identifier for each solution deployment
- **Timestamp**: Data-collection timestamp
- **Resized Instances**: The number of instances resized
- **Original Instance Size:** The original size of the instance
- **New Instance Size:** The new size of the instance

Note that AWS will own the data gathered via this survey. Data collection will be subject to the [AWS Privacy Policy](#). To opt out of this feature, modify the AWS CloudFormation template mapping section as follows:

```json
"Mappings": {
  "Send": {
    "AnonymousUsage": {
      "Data": "Yes"
    }
  },

  "Mappings": {
    "Send": {
      "AnonymousUsage": {
        "Data": "No"
      }
    }
  }
```
### Document Revisions

<table>
<thead>
<tr>
<th>Date</th>
<th>Change</th>
<th>In sections</th>
</tr>
</thead>
<tbody>
<tr>
<td>March 2019</td>
<td>Initial release</td>
<td>--</td>
</tr>
</tbody>
</table>

### Notices

Customers are responsible for making their own independent assessment of the information in this document. This document: (a) is for informational purposes only, (b) represents AWS's current product offerings and practices, which are subject to change without notice, and (c) does not create any commitments or assurances from AWS and its affiliates, suppliers or licensors. AWS’s products or services are provided “as is” without warranties, representations, or conditions of any kind, whether express or implied. AWS’s responsibilities and liabilities to its customers are controlled by AWS agreements, and this document is not part of, nor does it modify, any agreement between AWS and its customers.

Vertical scaling for the AWS Ops Automator is licensed under the terms of the Amazon Software License available at [https://aws.amazon.com/asl/](https://aws.amazon.com/asl/).

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.