

# EBS Snapshot Scheduler

## AWS Implementation Guide

*Puneet Agarwal*

*Lalit Grover*

*October 2016*

**Notice: EBS Snapshot Scheduler has been superseded by AWS Ops Automator.**

In 2016, the EBS Snapshot Scheduler was launched to help AWS customers automatically create snapshots of their Amazon Elastic Block Store (Amazon EBS) volumes on a defined schedule. In 2017, AWS launched the [AWS Ops Automator](#), a new and improved solution that enables customers to schedule EBS and Amazon Redshift snapshots, and automate other operational tasks. We encourage customers to migrate to AWS Ops Automator for future updates and new features.

Legacy templates, scripts, and documentation for EBS Snapshot Scheduler are available in our [GitHub repository](#).



Copyright (c) 2016 by Amazon.com, Inc. or its affiliates.

EBS Snapshot Scheduler is licensed under the terms of the Amazon Software License available at

<https://aws.amazon.com/asl/>

## Contents

Overview .....	3
Cost.....	3
Architecture Overview.....	4
Design Considerations.....	5
Real-Time Snapshots .....	5
Weekly Functionality .....	5
Time Zone Functionality .....	6
Regional Deployment.....	6
AWS CloudFormation Template .....	6
Automated Deployment .....	6
What We'll Cover.....	7
Step 1. Launch the Stack .....	7
Step 2. Tag Your Amazon EC2 Instances.....	9
Setting the Tag Value.....	10
Applying Custom Parameters.....	10
Modifying Tag Keys .....	11
Additional Resources.....	12
Appendix: Collection of Anonymous Data .....	13
Send Us Feedback.....	14
Document Revisions.....	14

## About This Guide

This implementation guide discusses architectural considerations and configuration steps for deploying the EBS Snapshot Scheduler on the Amazon Web Services (AWS) Cloud. It includes links to [AWS CloudFormation](#) templates that launch, configure, and run the AWS compute, network, storage, and other services required to deploy this solution on AWS, 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

Amazon Elastic Block Store (Amazon EBS) offers customers the ability to save point-in-time snapshots of their volumes. Amazon EBS snapshots can be used to instantiate multiple new volumes, expand the size of a volume, or move volumes across Regions and Availability Zones.

This guide provides infrastructure and configuration information for planning and deploying an EBS Snapshot Scheduler, which is a solution that automates the creation and optional deletion of EBS snapshots.

EBS Snapshot Scheduler leverages Amazon Elastic Compute Cloud (Amazon EC2) resource tags and AWS Lambda to automatically create EBS snapshots on a customer-defined schedule. The solution is easy to deploy and provides automated backups for data recovery and protection. For example, an organization can use the EBS Snapshot Scheduler in a production environment to automatically create EBS snapshots on daily basis, and schedule those snapshots to be deleted after a user-defined retention period. For customers who do not have a mechanism for taking regular EBS snapshots, this solution provides an easy way to improve data protection and recoverability.

## Cost

You are responsible for the cost of the AWS services used while running this solution. There is no additional cost for deploying the automated solution. The total cost of running this solution depends on the interval of the AWS Lambda function, the amount of space your initial snapshot consumes in Amazon Simple Storage Solution (Amazon S3), and the size of the changed blocks for each incremental snapshot. We recommend that you carefully consider your snapshot frequency and retention settings to avoid incurring unnecessary EBS snapshot charges.

As of the date of publication, the monthly cost for running this solution with default settings in US East (N. Virginia) is **approximately \$1.85** (\$1.27 for AWS Lambda, \$0.58 for Amazon DynamoDB) or less if you have free tier<sup>1</sup> monthly usage credit. The monthly cost does not include data transfer fees or EBS snapshot storage costs. Prices are subject to

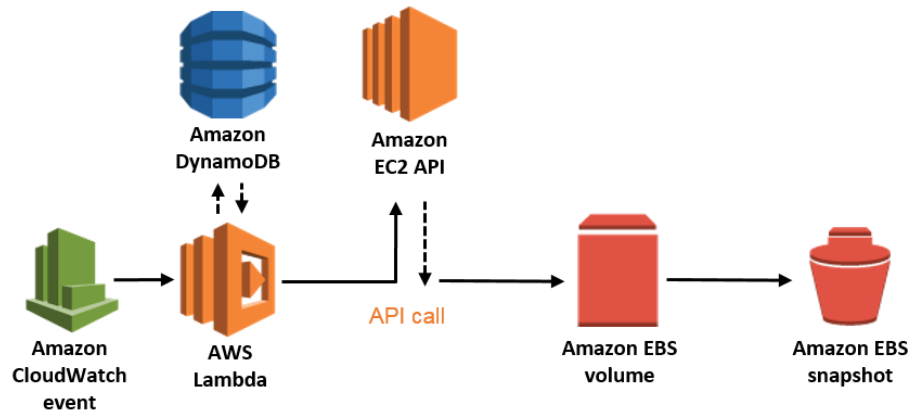
---

<sup>1</sup> <https://aws.amazon.com/lambda/pricing/>

change. For full details, see the pricing webpage for each AWS service you will be using in this solution.

## Architecture Overview

Deploying this solution with the **default parameters** builds the following environment in the AWS Cloud.



**Figure 1: EBS Snapshot Scheduler on AWS**

The AWS CloudFormation template sets up an Amazon CloudWatch event that runs every five minutes, by default. This event invokes the EBS Snapshot Scheduler AWS Lambda function. During configuration, the user defines a default snapshot time and time zone (UTC will be used if a time zone is not specified) as well as a *custom tag* that the EBS Snapshot Scheduler will use to identify applicable Amazon EC2 instances, and all EBS volumes attached to those EC2 instances. These configuration values are stored in Amazon DynamoDB, and the Lambda function retrieves them each time it runs. The customer then applies the custom tag to applicable Amazon EC2 instances and the EBS Snapshot Scheduler takes snapshots of all volumes attached to those instances.

For example, a user might assign the custom tag the name (tag key) `scheduler:ebs-snapshot`, and define a default snapshot time of 2030 UTC (8:30 PM) to be applied Monday through Thursday. In this example, the EBS Snapshot Scheduler runs every five minutes, but it only takes snapshots of tagged instances at 2030 UTC, Monday through Thursday.

**Note:** The Lambda function is invoked at five-minute intervals. If the default snapshot time is set between these intervals, the snapshot will be taken at the next Lambda invocation. For example, if the default snapshot time is set to 2032 UTC, and the Lambda runs at 2030 UTC and 2035 UTC, the snapshot will be taken at 2035 UTC.

Note that this solution allows the user to define custom snapshot time and retention parameters for an individual instance that override the default values. Users can also append additional characters to the custom tag name (e.g. `scheduler:ebs-snapshot:group1`) to easily create multiple snapshot schedules from the same EBS Snapshot Scheduler deployment. See [Step 2. Tag Your Amazon EC2 Instances](#) for detailed information about tag names and using the tag value to set custom parameters.

The solution includes an optional feature to automatically delete the EBS snapshots after a defined period. The default retention period is 15 days, but users can modify this default during configuration, and also adjust retention periods for individual instances (see [Applying Custom Parameters](#)). When this feature is enabled, the solution will apply a tag to applicable EBS snapshots to flag them for deletion.

When the EBS Snapshot Scheduler AWS Lambda function identifies an Amazon EC2 instance with a `scheduler:ebs-snapshot` tag, it will check if the tag value contains custom override parameters for the snapshot time or retention days. If there are no custom parameters, the Lambda function will initiate the snapshot and mark that snapshot for deletion according to the default settings.

**Important:** The EBS Snapshot Scheduler will create and delete snapshots for EBS volumes that are attached to both running and stopped EC2 instances.

## Design Considerations

### Real-Time Snapshots

The EBS Snapshot Scheduler takes snapshots in real time while the volume is attached to an Amazon EC2 instance and in use. Note that snapshots only capture data that has been written to your Amazon EBS volume, which might exclude any data that has been locally cached by your application or OS. We do not recommend using this solution to back up instances that require you to pause write files or unmount a volume to ensure a consistent and recoverable snapshot.

### Weekly Functionality

The current version of the EBS Snapshot Scheduler allows customers to create weekly schedules. For example, you can set it to take snapshots on specific days of the week or all days, however the current version does not support biweekly or monthly intervals. If you want to temporarily stop or modify EBS Snapshot Scheduler actions for a particular week or month, you must manually modify or remove resource tags on applicable Amazon EC2 instances.

## Time Zone Functionality

EBS Snapshot Scheduler allows customers to select a time zone for their snapshot schedules. When you specify a snapshot time and time zone, all snapshots are taken at that time in that time zone, regardless of the local time of the tagged instance. For example, if you set this solution to take snapshots at 8:00 PM ET, snapshots of tagged instances in the US West (Oregon) Region will be taken at 5:00 PM local time. If you want to take snapshots in the local time zone of the tagged instance, you must manually apply custom parameters to the instance, or group of instances. For more information, see [Step 2. Tag Your Amazon EC2 Instances](#).

## Regional Deployment

Customers can deploy the EBS Snapshot Scheduler in any AWS Region that supports AWS Lambda.<sup>2</sup> Once deployed, the EBS Snapshot Scheduler applies the appropriate snapshot actions to tagged Amazon EC2 instances in all AWS Regions<sup>3</sup> of a customer's account.

**Important:** EBS Snapshot Scheduler actions will affect appropriately tagged instances in all AWS Regions<sup>3</sup> of your account, even though the Lambda function is running in a single region.

## AWS CloudFormation Template

This solution uses AWS CloudFormation to bootstrap AWS infrastructure and automate the deployment of the EBS Snapshot Scheduler on the AWS Cloud. It includes the following CloudFormation templates, which you can download before deployment:

[View template](#)

**ebs-snapshot-scheduler.template:** This is the primary solution template you use to launch the EBS Snapshot Scheduler and all associated components. The default configuration deploys AWS Lambda functions, Amazon DynamoDB tables and an Amazon CloudWatch event, but you can also customize the template based on your specific needs.

## Automated Deployment

Before you launch the automated deployment, please review the architecture, configuration, and other considerations discussed in this guide. Follow the step-by-step

---

<sup>2</sup> For the most current AWS Lambda availability by region, see <https://aws.amazon.com/about-aws/global-infrastructure/regional-product-services/>

<sup>3</sup> At the time of publication, this does not include AWS GovCloud (US) or the China (Beijing) Region.

instructions in this section to configure and deploy the EBS Snapshot Scheduler into your account.

**Time to deploy:** Approximately five (5) 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 Stack](#)

- Launch the AWS CloudFormation template into your AWS account.
- Enter values for required parameters: **Stack name**.
- Review the other template parameters, and adjust if necessary.

### [Step 2. Tag Your Amazon EC2 Instances](#)

- Apply the custom tag to applicable instances.

## Step 1. Launch the Stack

This automated AWS CloudFormation template deploys EBS Snapshot Scheduler in AWS Lambda. Please make sure that you've verified the settings for your Amazon EC2 instances before launching the stack.

**Note:** You are responsible for the cost of the AWS services used while running this solution. See the [Cost](#) section for more details. 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 select the button to the right to launch the *ebs-snapshot-scheduler* 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 EBS Snapshot Scheduler in a different AWS Region, use the region selector in the console navigation bar.

A blue rectangular button with rounded corners containing the text "Launch Solution" in white.

**Note:** This solution uses the AWS Lambda service, which is currently available in specific AWS Regions only. Therefore, you must launch this solution in an AWS Region where Lambda is available.<sup>4</sup>

3. On the **Select Template** page, verify that you selected the correct template and choose **Next**.
4. On the **Specify Details** page, assign a name to your EBS Snapshot Scheduler stack.
5. Under **Parameters**, review the parameters for the template and modify them as necessary.

**Note:** The Lambda<sup>5</sup> function checks your current preferences in Amazon DynamoDB each time it runs, so you can manually modify these values at any time.

This solution uses the following default values.

Parameter	Default	Description
<b>CustomTagName</b>	scheduler:ebs-snapshot <Or user defined>	This tag identifies instances to receive automated actions, and also allows for custom snapshot time and retention days parameters. If you choose to modify the default value, make sure to assign a name that will be easy to apply consistently and correctly across all necessary instances.
<b>DefaultSnapshotTime</b>	2330	Default snapshot time. <b>Important:</b> All times are in 24-hour format, no colon.
<b>AutoSnapshotDeletion</b>	No	Enable automatic deletion of EBS snapshots after the retention period. If this parameter is set to <i>Yes</i> , snapshots will be deleted automatically after the number of days specified in <b>DefaultRetentionDays</b> or the custom tag value applied to the EC2 instance.
<b>DefaultRetentionDays</b>	15	Default snapshot retention period (number of days). Snapshots will be deleted only if you set <b>AutoSnapshotDeletion</b> to <i>Yes</i> .
<b>DefaultDaysActive</b>	All	Default days the EBS Snapshot Scheduler takes snapshots
<b>DefaultTimeZone</b>	UTC	Default time zone
<b>PolicyDDBTableName</b>	Scheduler-EBS-Snapshot-Policy	Name of the Amazon DynamoDB table where default snapshot settings are stored.

<sup>4</sup> For the most current AWS Lambda availability by region, see <https://aws.amazon.com/about-aws/global-infrastructure/regional-product-services/>

<sup>5</sup> All calls made to AWS Lambda must complete execution within 300 seconds. For large workloads, we recommend increasing the amount of memory for the EBS Snapshot Scheduler Lambda function.



Parameter	Default	Description
<b>HistoryDDBTableName</b>	Scheduler-EBS-Snapshot-History	Name of the Amazon DynamoDB table where snapshot history and deletion times are stored.
<b>ReadCapacityUnits</b>	1	Provisioned read throughput for DynamoDB tables.
<b>WriteCapacityUnits</b>	1	Provisioned write throughput for DynamoDB tables.
<b>SendAnonymousData</b>	Yes	Send anonymous data to AWS to help us understand EBS Snapshot Scheduler usage and related cost savings across our customer base as a whole. To opt out of this feature, choose No. For more information, see the appendix.

6. Choose **Next**.
7. On the **Options** page, choose **Next**.
8. On the **Review** page, review and confirm the settings. Be sure to check the box 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 roughly five (5) minutes.

**Note:** In addition to the primary AWS Lambda function `ebsSnapshotLambda`, this solution includes the `SolutionHelper` Lambda function, which runs only during initial configuration or when resources are updated or deleted.

When running this solution, you will see both Lambda functions in the AWS Lambda console, but only the primary `ebsSnapshotLambda` function is regularly active. However, do not delete the `SolutionHelper` function as it is necessary to manage associated resources.

## Step 2. Tag Your Amazon EC2 Instances

For the EBS Snapshot Scheduler to recognize an Amazon EC2 instance, the tag key on that Amazon EC2 instance must match (or start with) the custom tag defined when the solution was deployed. Therefore, it is important that you apply tags consistently and correctly to all applicable Amazon EC2 instances. You can continue to use existing tagging strategies for your instances while using this solution (see [Tagging Your Amazon EC2 Resources](#) for more information on instance tagging).

**Note:** After initial deployment, you can manually change the custom tag key in DynamoDB if necessary.

On the AWS Management Console, use the [Tag Editor](#) to apply or modify tags for multiple Amazon EC2 instances at a time. You can also apply and modify tags manually in the Amazon EC2 console.

## Setting the Tag Value

As you apply a tag, set the tag value accordingly to activate different EBS Snapshot Scheduler actions:

- To apply the default actions to an instance, set the tag value to `true` or `default`. The EBS Snapshot Scheduler will retrieve default parameters from the Amazon DynamoDB table.
- To temporarily exclude an instance from EBS Snapshot Scheduler actions, set the tag value to `none` or to an empty string (clear the field).
- To permanently exclude an instance from EBS Snapshot Scheduler actions, remove the solution's *custom tag*.
- To apply custom creation or retention parameters on an instance-by-instance basis, see the [next section](#).

## Applying Custom Parameters

You can apply custom creation and retention parameters to an instance which will override the default values you set during initial deployment. To do this, modify the tag value to specify the alternative settings.

The EBS Snapshot Scheduler will read tag values, looking for four possible custom parameters in following order: `<snapshot time>; <retention days>; <time zone>; <active day(s)>`

You must separate each value with a semicolon. The following table gives acceptable input values for each field.

Tag Value Field	Acceptable input values
<code>&lt;snapshot time&gt;</code>	Time in 24-hour format (with no colon)
<code>&lt;retention days&gt;</code>	Number of days (positive integer) to retain the snapshot before deletion, if set to automatically delete snapshots.

Tag Value Field	Acceptable input values
<b>Important:</b> To use this parameter, <b>AutoSnapshotDeletion</b> must be set to Yes.	
<time zone>	Time zone
<active day(s)>	all, weekdays, or mon, tue, wed, thu, fri, sat, and/or sun.

The following table gives examples of different tag values and the resulting EBS Snapshot Scheduler actions.

Example Tag Value	EC2 Scheduler Action
0800;6;utc;all	EBS snapshot will be taken at 0800 hours on all days of the week. Each snapshot will be retained for six days.
1000;2; us/eastern;weekdays	EBS snapshot will be taken at 1000 hours ET Monday through Friday. Each snapshot will be retained for two days.
1030;15; us/pacific;mon,tue,fri	EBS snapshot will be taken at 1030 hours PT on Monday, Tuesday, and Friday only. Each snapshot will be retained for 15 days.
0800	EBS snapshot will be taken at 0800 hours UTC on the default active days stored in the Amazon DynamoDB table. Each snapshot will be retained the default retention days stored in the DynamoDB table.
0800;5	EBS snapshot will be taken at 0800 hours UTC on the default active days stored in the Amazon DynamoDB table. Each snapshot will be retained for five days.
0800;5;us/pacific	EBS snapshot will be taken at 0800 hours PT on the default active days stored in the Amazon DynamoDB table. Each snapshot will be retained for five days.
default	EBS snapshots will be taken and retained on the default schedule.
True	EBS snapshots will be taken and deleted on the default schedule.
<EMPTY>	There will be no action taken on the instance.
<Random String>	There will be no action taken on the instance.

## Modifying Tag Keys

During initial deployment, the name you assign to the custom tag is stored in Amazon DynamoDB. When the AWS Lambda function runs, it will look for all instance tags that contain or start with that exact name. After it reads the name, it ignores any other characters on the tag key string.

As mentioned in the [Architecture Overview](#) section, you have the option to append additional characters to the tag key. This feature enables you to apply multiple sets of custom parameters to the same instance (or group of instances) using a single EBS Snapshot Scheduler deployment.

For example, a team deploys the solution using the custom tag key `scheduler:ebs-snapshot` and default parameters of a 1200 snapshot time (UTC) and 10 retention days. The following example scenario explains how they would modify the tags.

**Example:** There is a subset of instances that require three different snapshot schedules within a single week: an early snapshot on Monday and Wednesday, a mid-day snapshot Tuesday, Thursday, and Friday, and a late snapshot on weekends. Instead of deploying the full solution three times with three different default settings, the team uses a single deployment and applies three tags to those instances.

Tag Key	Tag Value
<code>scheduler:ebs-snapshot:early</code>	<code>0800;10;utc;mon,wed</code>
<code>scheduler:ebs-snapshot:mid</code>	<code>1200;10;utc;tue,thu,fri</code>
<code>scheduler:ebs-snapshot:late</code>	<code>2000;10;utc;sat,sun</code>

## Additional Resources

### AWS services documentation

- [AWS CloudFormation](#)
- [Amazon EBS Snapshots](#)
- [AWS Lambda](#)
- [Amazon DynamoDB](#)

## Appendix: Collection of Anonymous Data

This solution includes an option to send anonymous usage data to AWS. We use this data to better understand how customers use this solution to improve the services and products that we offer. When enabled, the following information is collected and sent to AWS each time the EBS Snapshot Scheduler AWS Lambda function runs:

- **Solution ID:** The AWS solution identifier
- **Unique ID (UUID):** Randomly generated, unique identifier for each EBS Snapshot Scheduler deployment
- **Timestamp:** Data-collection timestamp
- **EBS Snapshot Data:** Count of the EBS snapshots that are managed by the EBS Snapshot Scheduler in each AWS Region

Example data:

```
"us-west-2":{"snapshots_created":2,"snapshots_deleted":0,
"snapshots_existing": 10}
```

```
"us-east-1":{"snapshots_created":20,"snapshots_deleted":5,
"snapshots_existing": 55}
```

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, set the **SendAnonymousData** parameter to No.

# Send Us Feedback

We welcome your questions and comments. Please post your feedback on the [AWS Solutions Discussion Forum](#).

You can visit our [GitHub repository](#) to download the templates and scripts for this solution, and to share your customizations with others.

## Document Revisions

Date	Change	In sections
October 2016	Initial release	--

© 2016, Amazon Web Services, Inc. or its affiliates. All rights reserved.

### **Notices**

This document is provided for informational purposes only. It represents AWS's current product offerings and practices as of the date of issue of this document, which are subject to change without notice. Customers are responsible for making their own independent assessment of the information in this document and any use of AWS's products or services, each of which is provided "as is" without warranty of any kind, whether express or implied. This document does not create any warranties, representations, contractual commitments, conditions or assurances from AWS, its affiliates, suppliers or licensors. The responsibilities and liabilities of AWS 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.

The EBS Snapshot Scheduler solution is licensed under the terms of the Amazon Software License available at <https://aws.amazon.com/asl/>.