Cost Optimization Monitor

AWS Implementation Guide

Rafael Koike

Arthur Basbaum

Garvit Singh

Bryan Miller

November 2016

Contents

About This Guide

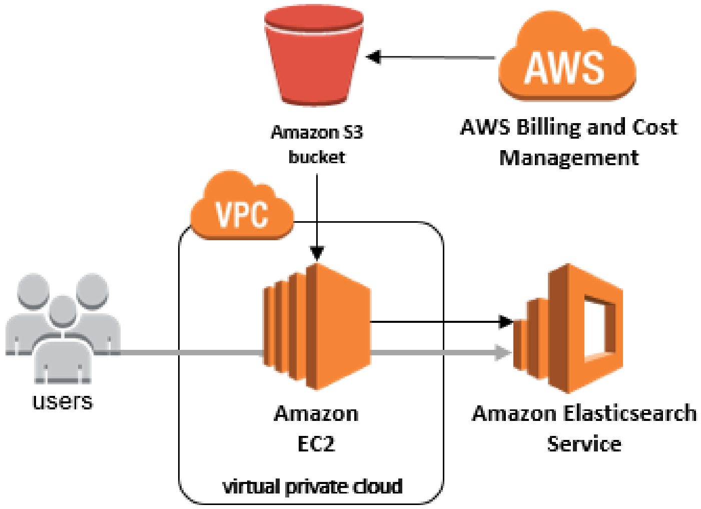
This implementation guide discusses architectural considerations and configuration steps for deploying the Cost Optimization Monitor on the Amazon Web Services (AWS) Cloud. It includes links to AWS CloudFormation templates that launch, configure, and run the AWS compute, network, storage, and other services 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 Web Services (AWS) enables customers to generate reports to gain insight into service usage and costs as they deploy and operate cloud architectures. These include detailed billing reports, which customers can access in the AWS Billing and Cost Management console. These reports provide estimated costs that customers can break down in different ways (by time period, account, resource, or custom resource tags) in order to help monitor and forecast their monthly charges. Organizations can analyze this information to help optimize their infrastructure and maximize their return on investment.

This guide provides infrastructure and configuration information for planning and deploying the Cost Optimization Monitor, which helps customers analyze their billing report data using Amazon Elasticsearch Service (Amazon ES) and its extended search, visualization, dashboard, and reporting capabilities. This AWS-provided, automated solution leverages AWS managed services, enabling deployment in a highly available and affordable way.



The information in this guide assumes basic knowledge of web, application, and operating system log formats. It is also helpful to have a working knowledge of Amazon Elasticsearch Service (Amazon ES) and Kibana for creating and customizing your own visualizations and dashboards.

Cost

You are responsible for the cost of the AWS services used while running this reference deployment. As of the date of publication, the cost for running the Cost Optimization Monitor with default settings in the US East (N. Virginia) Region is as shown in the table below. This includes charges for Amazon Elasticsearch instance hours and storage, Amazon EC2 usage and licensing, and Elastic Load Balancing base pricing.

Deployment Size	Amazon ES Master Node Count and Type	Amazon ES Data Node Count and Type	Total Cost/Hour
Small	3 - t2.small.elasticsearch	2 - m3.medium.elasticsearch	$0.40
Medium	3 - t2.small.elasticsearch	4 - m3.large.elasticsearch	$1.13
Large	3 - m3.medium.elasticsearch	8 - m3.xlarge.elasticsearch	$3.83

This pricing does not reflect variable charges incurred from Amazon Simple Storage Service (Amazon S3) and Elastic Load Balancing data throughput. 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.

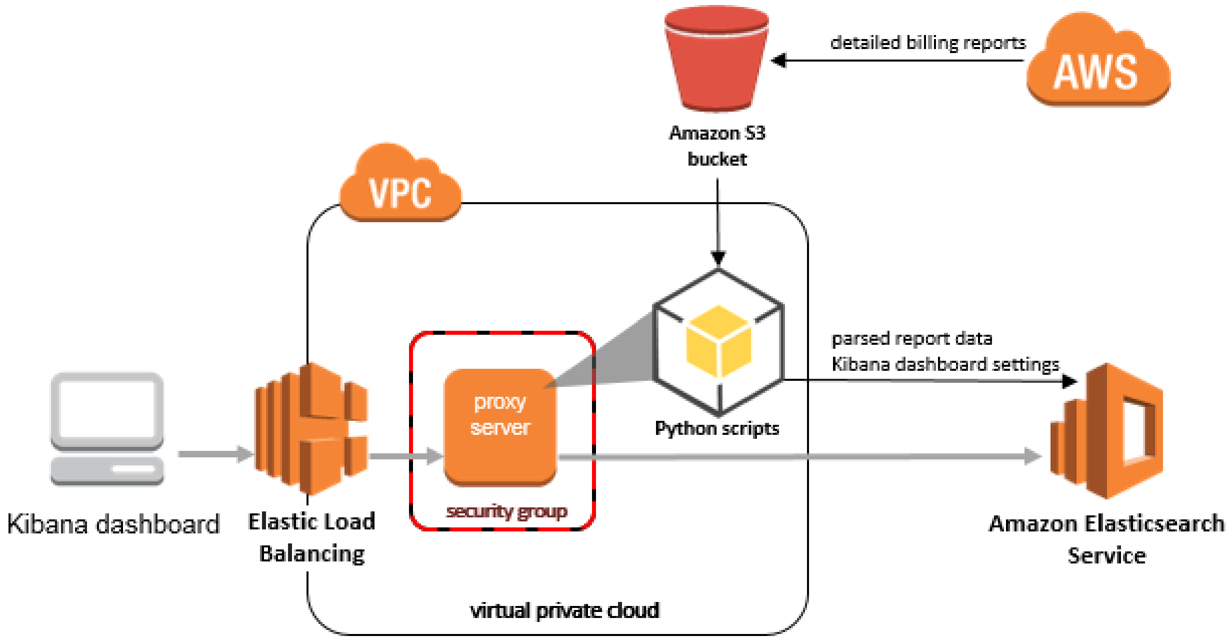


Figure 1: Cost Optimization Monitor solution architecture on AWS

The solution deploys an Amazon Elasticsearch Service (Amazon ES) domain, which is the hardware, software, and data exposed by Amazon Elasticsearch Service endpoints. During initial configuration of the AWS CloudFormation template, users choose from one of three solution sizes to determine the number and type of data nodes (Amazon Elasticsearch instances) in the cluster: small, medium, or large.

The template also launches an Amazon EC2 instance in an Amazon Virtual Private Cloud (Amazon VPC) network. The instance is configured with an Nginx proxy to limit the exposure of data stored in Amazon ES. It serves as an intermediary between the Kibana client web browser and the Amazon ES domain endpoint, filtering requests and then forwarding them to Amazon ES from a single authenticated IP address (see Additional Security Settings for more information). During initial configuration, the user specifies the IP address range that can access the instance as well as custom login credentials for an extra layer of protection. This highly available design enables Auto Scaling to maintain solution performance and uses Elastic Load Balancing to manage incoming traffic.

The Amazon EC2 instance also hosts Python scripts that parse data from the latest billing report and upload it daily to Amazon ES, and also that import a default set of Kibana dashboards to Amazon ES during initial launch.

Users have the option to automatically create a new Amazon S3 bucket or use an existing bucket to store detailed billing reports. The template configures the applicable bucket with a cross-account policy to allow objects from AWS Billing and Cost Management. The report file is continuously overwritten, but the data will be parsed daily and sent to Amazon ES.

Solution Features

The automated Cost Optimization Monitor solution provides the following features:

- **High Availability –** Amazon ES is a managed service that automatically replicates and distributes your multi-node Elasticsearch cluster across different Availability Zones. This solution uses Auto Scaling and Elastic Load Balancing to maintain availability of the proxy server, which manages all client requests to the Amazon ES endpoint.
- **Custom Sizing –** Choose from three preset Amazon ES cluster sizes to support your anticipated report data:

  Small:
  - 3 dedicated master nodes; t2.small.elasticsearch instance type
  - 2 data nodes; m3.medium.elasticsearch instance type

  Medium:
  - 3 dedicated master nodes; t2.small.elasticsearch instance type
  - 4 data nodes; m3.large.elasticsearch instance type

  Large:
  - 3 dedicated master nodes; m3.medium.elasticsearch instance type
  - 8 data nodes; m3.xlarge.elasticsearch instance type
- **Security –** This solution uses an Nginx proxy server to restrict access to the Kibana dashboard. It also creates a security group that gives you fine-grained control over access to that proxy server.
- **Scalability –** Modify your cluster's instance count and type directly in Amazon ES to accommodate your changing environment and requirements, without having to reconfigure the solution architecture or manage backend resources. Auto Scaling is enabled on the Amazon EC2 instance to help ensure you meet future demand.
- **Custom Reporting –** Take advantage of Kibana 4 features to create, save, and share custom visualizations and customer views. This solution includes a configuration file to get you started with some popular dashboard views.

AWS CloudFormation Template

This solution uses AWS CloudFormation to automate the deployment of the Cost Optimization Monitor on the AWS Cloud. It includes the following AWS CloudFormation template, which you can download before deployment:

View template

cost-optimization-monitor.template: Use this template to launch the Cost Optimization Monitor solution and all associated components, as described in the previous section. The default configuration offers three deployment size options, 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 security information discussed in this guide. Follow the step-by-step instructions in this section to configure and deploy the Cost Optimization Monitor into your account.

Time to deploy: Approximately 25 minutes

Prerequisites

Customers who use Consolidated Billing to manage payments for multiple accounts must launch this template in the payer account. For more information on Consolidated Billing, see the AWS Billing and Cost Management User Guide.

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**, **User Name**, **Password, Access CIDR Block, SSH Key, Domain Name**.
- Review the other template parameters, and adjust if necessary.

Step 2. Configure Billing Reports to Save to Amazon S3

- Configure detailed billing reports to save to the appropriate Amazon S3 bucket.

Step 3. Open the Kibana Dashboard

- View the first batch of billing data in the preconfigured dashboard.

Step 1. Launch the Stack

This automated AWS CloudFormation template deploys the Cost Optimization Monitor on the AWS Cloud.

> **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 click the button to the right to launch the *cost-optimization-monitor* AWS CloudFormation template.

   Launch Solution

   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 Cost Optimization Monitor solution in a different AWS Region, use the region selector in the console navigation bar.

   > **Note**: During initial deployment, this solution uses AWS Lambda, which is currently available in specific AWS Regions only. Therefore, you must launch this solution an AWS Region where Lambda is available. [1]

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 Cost Optimization Monitor stack.

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

Parameter	Default	Description
User Name	<Requires input>	User name for access to the Amazon EC2 instance
Password	<Requires input>	Password for access to the Amazon EC2 instance **Note:** Must be six characters or longer and must contain one uppercase letter, one lower case letter, and a special character (!@#$%^&+)
Access CIDR Block	<Requires input>	This IP address range will have HTTP and SSH access to the Amazon EC2 instance.

[1] For the most current service availability by AWS Region, see https://aws.amazon.com/about-aws/global-infrastructure/regional-product-services/

Parameter	Default	Description
SSH Key	<Requires input>	Public and private key pair, which allows you to connect securely to the Amazon EC2 instance. When you created an AWS account, this is the key pair you created in your preferred AWS Region.
Domain Name	<Requires input>	The name of the Amazon ES domain that this template will create. **Note:** Amazon ES domain names must start with a lowercase letter and must be between 3 and 28 characters. Valid characters are a-z (lowercase only), 0-9, and – (hyphen).
Cluster Size	`Small`	A drop-down box with three options: `Small`, `Medium`, `Large`
Use Existing Bucket?	`No`	Choose `Yes` to use an existing Amazon S3 bucket. Otherwise, choose `No` to create one at launch.
Existing S3 Bucket Name	`<Requires input>`	If you chose `Yes` above, enter an existing bucket name. Otherwise, leave this field blank.
VPC CIDR Block	`10.255.0.0/16`	CIDR block for the solution's VPC. You can modify the address range to avoid overlapping with existing networks.
1st Subnet Network	`10.250.250.0/24`	CIDR block for the VPC subnet created in AZ1
2nd Subnet Network	`10.250.251.0/24`	CIDR block for the VPC subnet created in AZ2
Send Anonymous Usage Data	`Yes`	Send anonymous data to AWS to help us understand solution usage and related cost savings across our customer base as a whole. To opt out of this feature, select `No`. For more information, see Appendix B.

6. Choose **Next**.
7. On the **Options** page, you can specify tags (key-value pairs) for resources in your stack and set additional options, and then 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 20 minutes.
10. To see details for the stack resources, choose the **Outputs** tab. The following table describes each of these outputs in more detail.

Key	Description
BucketName	Amazon S3 bucket for storing detailed billing records
SingleDashboardURL	For single AWS accounts: the URL for front-end access to the Kibana 4 dashboard via the proxy server **Note:** Use this link to open the Kibana dashboard webpage. It will not have billing data to show immediately. You must wait for the first batch of detailed billing reports from AWS Billing and Cost Management, and then for the python scripts to parse and upload that data. This occurs daily as a scheduled cron job.
ConsolidatedDashboardURL	For AWS accounts using Consolidated Billing: the URL for front-end access to the Kibana 4 dashboard via the proxy server

Note: This solution deploys an AWS Lambda function, `solution-helper`, which runs only during initial configuration or when resources are updated or deleted. You will see the `solution-helper` function in the AWS Lambda console, which is necessary to manage associated resources for as long as the solution is running.

Step 2. Configure Billing Reports to Save to Amazon S3

After the Cost Optimization Monitor stack launch completes, complete the following procedure to configure your account to save detailed billing records to the Amazon S3 bucket that this solution created.

1. In the AWS CloudFormation console, in the **Outputs** tab, select and copy the name of the S3 bucket (the **BucketName** description).
2. Open the Billing and Cost Management console. (To navigate to the console, select **Billing & Cost Management** from the drop-down box field under your role in the top menu bar.)
3. On the left navigation pane, choose **Preferences.**
4. Select the **Receive Billing Reports** check box.
5. In the **Save to S3 Bucket** field, paste the bucket name and choose **Verify** to make sure the bucket policy has been configured with the correct permissions (the solution's AWS CloudFormation template configures the policy automatically).
6. Under **Report**, select the **Detailed billing report with resources and tags*** check box and choose **Save preferences**.

It can take up to 24 hours for AWS to start delivering detailed billing reports to your S3 bucket. Once they are delivered, they will be processed with the next scheduled run of the Python scripts. As more billing data is generated, the detailed billing report data is overwritten in the bucket.

Step 3. Open the Kibana Dashboard

A Kibana dashboard displays a group of visualizations that you can modify, save, and share. The sample visualizations for this solution display information such as cost by resource tag, Amazon EC2 instances running each hour, and the percentage of instances using the Reserved Instance or Spot instance pricing model. See Appendix A for detailed information on each cost metric in the default dashboard.

The solution uses a cron expression (cron job) to run the Python scripts and process the detailed billing report data daily at 2330 UTC (current month's data) and midnight UTC (previous month's data). As mentioned previously, it can take up to 24 hours to receive your first detailed billing reports. Once the reports are delivered to your S3 bucket, they will be processed with the next scheduled Python run. The solution automatically imports your data to Kibana and configures an initial dashboard. After your first batch of data is processed, you will be able to see sample visualizations in the Kibana dashboard.

1. Go to the AWS CloudFormation console, and in the **Outputs** tab, open the applicable link (**SingleDashboardURL** or **ConsolidatedDashboardURL**) to go to the Kibana dashboard.
2. When prompted, log in to the dashboard with the user name and password you specified in Step 1. Launch the Stack.
3. A default dashboard will load. In the upper-right corner, you can adjust the data time period (clock icon). You can also set an interval for the webpage refresh rate (**Auto-refresh**).
4. In the top menu under **Settings**, choose the **Objects** tab. You will see the default saved dashboards for this solution.

Explore and experiment with the dashboard settings to create and save additional visualizations. For more information, go to the Kibana User Guide.

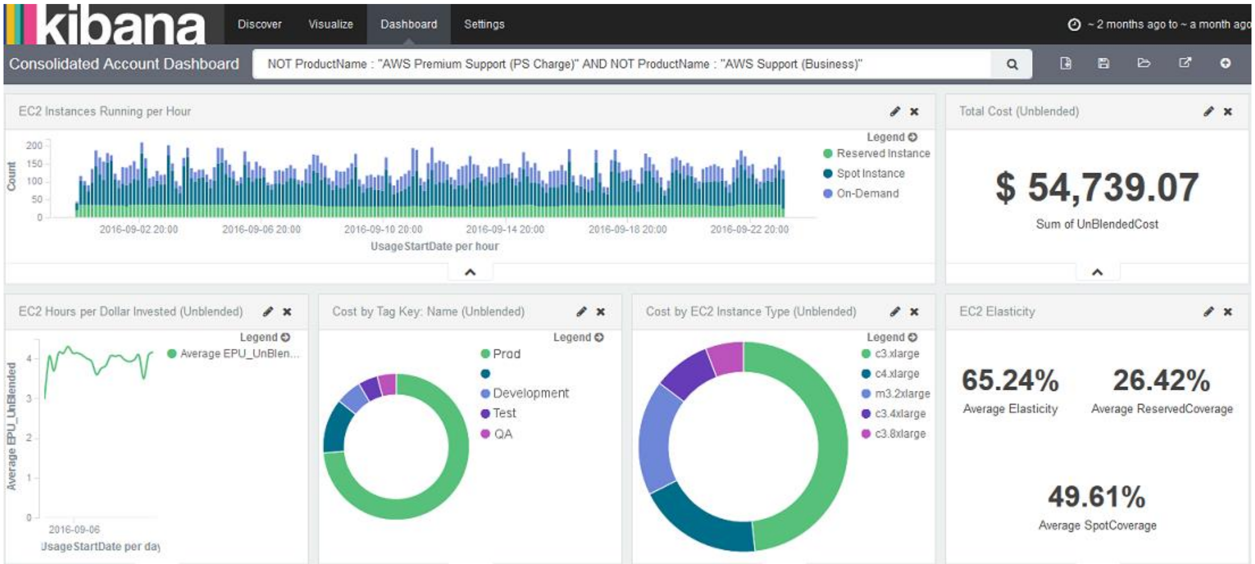


Figure 2: Example Kibana dashboard

Security

The AWS Cloud provides a scalable, highly reliable platform that helps customers deploy applications and data quickly and securely. When you build systems on AWS infrastructure, security responsibilities are shared between you and AWS. This shared model can reduce your operational burden as AWS operates, manages, and controls the components from the host operating system and virtualization layer down to the physical security of the facilities in which the services operate. In turn, you assume responsibility and management of the guest operating system (including updates and security patches), other associated applications, as well as the configuration of the AWS-provided security group firewall. For more information about security on AWS, visit the AWS Security Center.

Security Groups

The security groups created in this solution are designed to control and isolate network traffic between the Amazon EC2 instance and your Amazon ES domain. Traffic to port 22 and port 80 is restricted to the range specified in the **Access CIDR Block** AWS CloudFormation parameter. The security group for the Amazon ES domain is restricted to allow traffic from only the solution’s Amazon EC2 instance. We recommend that you review the security groups and adjust access settings as needed once the deployment is up and running.

Additional Security Settings

An Nginx proxy is added to the architecture to enable strict security controls and limit the exposure of data stored in Amazon ES. The proxy server acts as an intermediary between the Kibana client web browser and the Amazon ES domain endpoint, filtering requests and then forwarding them to Amazon ES from a single, authenticated IP address.

The proxy server uses two security mechanisms to handle inbound requests from Kibana: authentication (user name and password) and IP restriction (security group). When an end user attempts to access the domain dashboard, a login prompt appears. The Kibana client forwards the user name and password along with the requester's source IP address to the proxy server for evaluation. If the credentials match and the source IP address is within the approved range, the proxy server then passes the request to the Amazon ES endpoint. When the Amazon ES endpoint has responded, the proxy server returns that information to the client's web browser.

Note that Kibana is JavaScript based and, therefore, all requests that it forwards originate from unauthenticated end-user IP addresses. Customers can configure IP-based access policies from Amazon ES domain endpoints, however these endpoints require Signature Version 4 signing to grant access to the service. This makes it burdensome to manage requests from Kibana directly in Amazon ES, as customers would need manage a whitelist of individual IP addresses. The Nginx proxy server simplifies the management of inbound traffic while providing an authenticated, single origin for all requests to Amazon ES. For more information, see How to Control Access to Your Amazon Elasticsearch Service Domain in the AWS Security Blog.

Additional Resources

- AWS Cost Optimization
- Cost Optimization: EC2 Right Sizing
- AWS CloudFormation
- Amazon Elasticsearch Service
- Kibana User Guide
- Amazon EC2 user guide for Linux instances
- AWS SDK for Python (Boto)
- Amazon VPC
- Amazon S3
- AWS IAM

Appendix A: Default Dashboard Metrics

The solution's default dashboard is configured to show specific cost and usage metrics. We selected these metrics based on best practices observed across AWS customers.

All metrics are calculated using the **Detailed billing report with resource and tags** that is automatically generated from the Billing and Cost Management console. These calculations do not include data from other sources such as Amazon CloudWatch. We recommend that customers regularly monitor service usage and logs to make a comprehensive assessment of whether their resources are using the most appropriate and cost-effective for their technical workloads.

> **Note:** The **Consolidated Account Dashboard** shows unblended costs, which reflect actual costs per resource per account, not the average costs that are distributed across a linked account family. For detailed information, see Understanding Consolidated Bills.

EC2 Instances Running per Hour

- **Description:** This visualization shows the number of Amazon EC2 instances running per hour by purchase type: On-Demand, Reserved Instances, or Spot instances. Use this chart to help assess how elastic your environment is and identify opportunities to increase capacity reservations for additional cost savings.
- **Formula:** Count distinct running Amazon EC2 instances per hour and by purchase type
- **Unit:** Instance count

Total Cost

- **Description:** This shows the total monthly cost to date for a single account or unblended cost across all linked accounts.
- **Formula:** SUM (Total Cost)
- **Unit:** US dollars (USD)

EC2 Hours per Dollar Invested

- **Description:** This visualization shows how many hours of Amazon EC2 utilization you are getting per dollar ($1 USD) invested in AWS. This is a useful metric for trend analysis. For example, after moving some capacity from the On-Demand pricing model to Reserved Instances, this number should increase. Similarly, if you are outbid for Spot instances and forced to launch On-Demand instances, this number should decrease.

- **Formula:** 1 / AVG(SUM (EC2 Cost))
- **Unit:** Hours

Cost by Tag Key: Name

- **Description:** This visualization shows a cost breakdown by tag value for AWS resources with the commonly used tag key `Name`, which can help customers understand costs that correspond to specific tag groupings. You can modify this visualization to show data for a different tag key. A blank section in the chart represents blank tag values.
- **Formula:** Cost per tag value
- **Unit:** US dollars (USD)

Cost by EC2 Instance Type

- **Description:** This visualization breaks down your investment per Amazon EC2 instance type to help identify which instances are incurring the most cost.
- **Formula:** Cost per Amazon EC2 instance type
- **Unit:** US dollars (USD)

EC2 Elasticity

Average Elasticity

- **Description:** This metric shows the percentage of on-demand Amazon EC2 instances that you stop daily. This is a valuable metric because it shows how effectively your environment is balancing capacity and demand. Highly elastic environments (>30%) will get higher savings.
- **Formula:** (Daily-AVG-MAX(# EC2 On-Demand Instances) – Daily-AVG-MIN(# EC2 On-Demand Instances)) / Daily-AVG-MAX(# EC2 On-Demand Instances)
- **Unit:** Percentage

Average Reserved Instance/Spot Instance Coverage

- **Description:** These metrics show the percentage of instances running under the Reserved Instance and Spot instance purchase models. These instances can provide you with a significant discount compared to On-Demand instance pricing. A higher percentage in this field corresponds to cost-optimized environments.
- **Formula:** AVG(# RI EC2 Instances)/AVG(# Total EC2 Instances) and AVG(# Spot EC2 Instances)/AVG(# Total EC2 Instances)
- **Unit:** Percentage

Explore and experiment with the Kibana dashboard settings to create and save additional visualizations, and share your ideas in our GitHub repository.

Appendix B: 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:

- **Solution ID:** The AWS solution identifier
- **Unique ID (UUID):** Randomly generated, unique identifier for each Cost Optimization Monitor deployment
- **Timestamp:** Data-collection timestamp
- **Cluster Size:** Size of the Amazon Elasticsearch cluster the solution will deploy

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 **Send Anonymous Usage Data** 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 the GitHub repository to download the templates and scripts for this solution, and to share your customizations with others.

Document Revisions

Date	Change	In sections
November 2016	Initial publication	-

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.