Centralized Logging
Implementation Guide
# Table of Contents

Welcome .......................................................... 1  
Cost ........................................................................ 2  
  Cost estimate example ........................................... 2  
Architecture overview ........................................... 4  
  Log ingestion ......................................................... 4  
  Log indexing ......................................................... 6  
  Visualization ......................................................... 6  
Security ....................................................................... 8  
  OpenSearch Service in Amazon VPC ......................... 8  
  Amazon Cognito ..................................................... 8  
  Access Policy ........................................................ 9  
Considerations ......................................................... 10  
  Custom sizing ....................................................... 10  
  Scalability ............................................................ 10  
  Deployment options .............................................. 10  
    Sample logs ....................................................... 10  
    Logging across accounts and Regions .................... 11  
Solution updates .................................................... 11  
Regional deployments ............................................. 11  
Templates ............................................................... 12  
Automated deployment ............................................ 13  
  Deployment overview ............................................ 13  
  Step 1. Launch the stack ......................................... 13  
  Step 2. (Optional) Configure the Kibana dashboard ........ 16  
    Import dashboard .............................................. 16  
  Step 3. (Optional) Launch the demo stack in your spoke accounts or spoke Regions .......... 18  
Additional resources ............................................... 20  
Sample logs ............................................................ 21  
  Sample logs Apache web server ............................... 21  
Accessing Jumpbox ................................................ 22  
Adding custom CloudWatch Logs ............................. 23  
Adding social identity providers ............................... 24  
Troubleshooting ..................................................... 26  
  Common Errors .................................................... 26  
    Validation error with the AWS CloudFormation primary template ..................... 26  
    Error in creating a CloudWatch Logs subscription filter ................................. 27  
    CloudWatch Logs group already exists ......................................................... 27  
    Invalid Region parameter entered in the primary template ......................... 28  
    An Amazon VPC limit error ....................................... 28  
    An AWS CloudFormation nested stack stays in a REVIEW_IN_PROGRESS status ...... 29  
Migrate your solution data ......................................... 30  
Uninstall the solution ............................................... 31  
  Using the AWS Management Console ....................... 31  
  Deleting the Amazon S3 buckets .............................. 31  
  Deleting the CloudWatch Logs ............................... 32  
  Deleting the Amazon OpenSearch Service domain .......... 32  
  Deleting the Amazon VPC ....................................... 32  
Using AWS Command Line Interface .......................... 32  
Operational metrics ................................................. 33  
Source code ........................................................... 34  
Revisions .................................................................... 35  
Contributors ............................................................ 36  
Notices ....................................................................... 37
Collect, analyze, and display Amazon CloudWatch Logs in a single dashboard with the Centralized Logging solution

Publication date: November 2016 (last update (p. 35): December 2021)

The Centralized Logging solution collects, analyzes, and displays Amazon CloudWatch Logs in a single dashboard. Amazon Web Services (AWS) services generate log data, such as audit logs for access, configuration changes, and billing events. In addition to AWS log data, web servers, applications, and operating systems all generate log files in various formats. Consolidating, managing, and analyzing these different log files is challenging to customers.

This solution contains a suite of infrastructure services that deploy a centralized logging solution. You can collect Amazon CloudWatch Logs from multiple accounts and AWS Regions. It uses Amazon OpenSearch Service and Kibana, an analytics and visualization platform that is integrated with Amazon OpenSearch Service, to create a unified view of all the log events. In combination with other AWS managed services, this solution provides customers with a turnkey environment to begin logging and analyzing your AWS environment and applications.

This solution also includes a demo AWS CloudFormation template that deploys sample logs, which you can use for testing purposes. We recommend deploying this optional template so that you can test how the solution works with sample logs generated by the demo resources.

The information in this guide assumes basic familiarity of web, application, and operating system log formats. Working knowledge of Amazon OpenSearch Service and Kibana for creating and customizing your own dashboards and visualizations, is recommended.

This implementation guide describes architectural considerations and configuration steps for deploying the Centralized Logging solution in the AWS Cloud. It includes links to 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 in the AWS Cloud.
Cost

You are responsible for the cost of the AWS services used while running this solution. As of December 2021, the cost for running this solution using a small Amazon OpenSearch Service cluster size, and assuming an average record size of 5 KB at a rate of 1 records/second for Amazon Kinesis Data Firehose and 1 shard provisioned for Amazon Kinesis Data Streams, with default settings in the US East (N. Virginia) Region is approximately $832.08/month. A detailed breakdown of this cost estimate is provided in the following tables.

Cost estimate example

The cost to run Amazon OpenSearch Service with different cluster sizes is shown in the following table, which includes charges for Amazon OpenSearch Service instance hours.

<table>
<thead>
<tr>
<th>OpenSearch Service cluster size</th>
<th>Total cost/month</th>
</tr>
</thead>
<tbody>
<tr>
<td>Small</td>
<td>$806.40</td>
</tr>
<tr>
<td>Medium</td>
<td>$3,477.46</td>
</tr>
<tr>
<td>Large</td>
<td>$6,696.00</td>
</tr>
</tbody>
</table>

The following example cost estimate uses a small Amazon OpenSearch Service cluster size as the basis for calculation and assumes an average record size of 5 KB at a rate of 1 record/second. To determine the cost estimate using different Amazon OpenSearch Service cluster sizes, replace the Total Cost amount with the cost for the corresponding cluster size.

Note
The cost for running this solution is a formula based on size of your log records and the rate at which this data is being sent to the Kinesis Data and Delivery Streams. Transfer costs between Regions using CloudWatch destinations are not accounted for in this cost estimate.

<table>
<thead>
<tr>
<th>AWS service</th>
<th>Example use case</th>
<th>Cost per month</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amazon OpenSearch Service</td>
<td>Small cluster size</td>
<td>$806.40</td>
</tr>
</tbody>
</table>
| Amazon Kinesis Data Firehose | • Data ingestion (GB/second): (1 record/second x 5 KB/record) or approximately 12.359 GB/month  
• US East (N. Virginia) Region costs  
• Data ingestion: $0.029/GB for first 500 TB/month  
• Data processing to Amazon VPC: $0.01/GB  
• Amazon VPC delivery: $0.01 per hour, per Availability Zone for VPC delivery | $14.88         |
<table>
<thead>
<tr>
<th>AWS service</th>
<th>Example use case</th>
<th>Cost per month</th>
</tr>
</thead>
</table>
| Amazon Kinesis Data Streams | • Sample logs put approximately four to five records/minute with a data throughput of approximately 1 KB/second  
  • 1 shard ingests up to 1 MiB/second  
  • PUT payload unit (25 KB): one record is 1 PUT payload unit  
  • Aggregation: approximately five records/minute which equates to approximately 216,000 PUT payload units/month                                                                 | $10.80         |
|                             |                                                                                                                                                                                                                 |                |
|                             | **Total:**                                                                                                                                                                                                       | **$832.08**    |

This cost estimate does not reflect variable, usage-driven charges incurred from Amazon CloudWatch, AWS CloudTrail, and AWS Lambda. The two most significant cost contributors for this solution are Kinesis Data Firehose and Kinesis Data Streams. For information on Kinesis Data Firehose pricing and sample scenarios, refer to the Amazon Kinesis Data Firehose pricing page. For information on Kinesis Data Streams pricing and sample scenarios, refer to the Amazon Kinesis Data Streams pricing page. Prices are subject to change. For full details, refer to the pricing webpage for each AWS service used in this solution.
Architecture overview

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

![Centralized Logging solution architecture on AWS](image)

**Figure 1: Centralized Logging solution architecture on AWS**

The Centralized Logging solution contains the following components: log ingestion, log indexing, and visualization.

**Note**
Deploy the AWS CloudFormation template in the AWS account where you intend to store your log data.

Log ingestion

For the log ingestion component, the AWS CloudFormation template deploys Amazon CloudWatch Logs destinations in the primary account.
1. This solution uses the CloudWatch Logs destination capability for log streaming. CloudWatch Logs destinations are created with the required permissions in each of the selected Regions in your primary account. After the destinations are created with the necessary permissions, you can configure CloudWatch Logs subscription filters for log groups to be streamed to the centralized logging account. For information about creating custom CloudWatch Logs, refer to Adding custom CloudWatch Logs (p. 23).

   **Note**
   You can control the spoke streams that will log events to the primary account using the Spoke Account and Spoke Region parameters. These parameters can be updated at any time after installation to add/remove accounts and Regions.

2. An optional demo AWS CloudFormation template can be deployed to generate sample CloudWatch Logs for AWS CloudTrail, Amazon Virtual Private Cloud (Amazon VPC) flow logs, and an Amazon Elastic Compute Cloud (Amazon EC2) web server. For information about the sample logs, refer to Sample logs (p. 21). The demo template configures each of these log groups with the needed subscription filters to stream log events to the CloudWatch Logs destination in the centralized logging account, as shown in Figure 2.

   **Important**
   Since the sample logs Apache web server is publicly accessible, we do not recommend deploying the demo AWS CloudFormation template in a production environment.
Log indexing

For the log indexing component, the AWS CloudFormation template deploys Amazon Kinesis Data Streams, AWS Lambda functions, Amazon Kinesis Data Firehose, and Amazon OpenSearch Service, as shown in Figure 3.

1. A centralized Kinesis Data Streams is provisioned to index log events on the centralized Amazon OpenSearch Service domain. The CloudWatch Logs destinations created to stream log events, have Kinesis Data Streams as their target.

2. Once the log events stream to Kinesis Data Streams, the service invokes an AWS Lambda function to transform each log event to an Amazon OpenSearch Service document, which is then put in Kinesis Data Firehose.

3. Kinesis Data Firehose indexes the documents on the Amazon OpenSearch Service domain.

4. Kinesis Data Firehose logs errors in CloudWatch and delivers the records to Amazon Simple Storage Service (Amazon S3) for low-cost storage.

   **Note**
   
   You can monitor Kinesis Data Firehose as it sends custom CloudWatch Logs containing detailed monitoring data for each delivery stream.

Visualization

This solution provides data visualization and exploration support using Amazon OpenSearch Service and Kibana. An Amazon OpenSearch Service domain is created inside an Amazon VPC, preventing public access to the Kibana dashboard. Access to the Kibana dashboard is secured using a VPC security group and an AWS Identity and Access Management (IAM) role.
Figure 4: Centralized Logging visualization component

This solution optionally launches a Microsoft Windows Jumpbox Server that can be used to access the Amazon OpenSearch Service cluster and Kibana dashboard. An administrator account is configured to provide the permission to access the Kibana dashboard. This solution uses an Amazon Cognito user pool and an identity pool for authentication and authorization. For additional information about this solution's security, refer to Security (p. 8).
Security

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. For more information about security on AWS, visit AWS Cloud Security.

### Amazon OpenSearch Service in Amazon VPC

The centralized Amazon OpenSearch Service domain that is configured in this solution deploys in an Amazon VPC, which restricts access based on appropriate security group rules. Kibana cannot be accessed from outside the Amazon VPC. The solution optionally deploys a Windows Jumpbox Server to access the Kibana URL.

### Amazon Cognito

Amazon OpenSearch Service uses Amazon Cognito to provide secure access (username and password) to Kibana. An admin user is created in a Cognito user pool as part of the solution's deployment. Access
to the Amazon OpenSearch Service domain is restricted to authorized users using an Amazon Cognito identity pool. To activate federated access for social identity providers, refer to Adding social identity providers (p. 24). For information about enabling multi-factor authentication (MFA) to a user pool, refer to Adding Multi-Factor Authentication (MFA) to a User Pool in the Amazon Cognito Developer Guide.

### Access policy

The Centralized Logging solution creates an Amazon OpenSearch Service domain with an access policy that restricts access to two roles:

- **Kinesis Data Firehose role:** to allow Kinesis Data Firehose to index documents in the Amazon OpenSearch Service domain.
- **Authenticated users role:** to allow authorized users to view indexed data on the Kibana dashboard.
Design considerations

Custom sizing

This solution offers three preset Amazon OpenSearch Service cluster sizes to help support your anticipated log traffic:

- **Small:**
  - 3 dedicated primary nodes; c5.large.search instance type
  - 4 data nodes; r5.large.search instance type
- **Medium:**
  - 3 dedicated primary nodes; c5.large.search instance type
  - 6 data nodes; r5.2xlarge.search instance type
- **Large:**
  - 3 dedicated primary nodes; c5.large.search instance type
  - 6 data nodes; r5.4xlarge.search instance type

Scalability

You can modify your cluster’s instance count and type directly in Amazon OpenSearch Service to accommodate your changing environment and requirements, without having to reconfigure the solution’s architecture or manage backend resources. As a best practice, we recommend that you monitor your cluster’s performance metrics, which can help you update the cluster size for future business needs. Refer to the Amazon OpenSearch Service FAQs for information about Amazon OpenSearch Service clusters.

Deployment options

This solution offers the following deployment options:

1. Deploy the primary template into your AWS account without enabling sample logs; toggle the **Sample Log** parameter to No.
2. Deploy the primary template into your AWS account with sample logs activated, which sets up a single region within the primary account for generating demo data; toggle the **Sample Log** parameter to Yes.
3. Deploy the primary template into your AWS account and deploy the demo template into spoke accounts or spoke Regions to test the multi-account, multi-Region set up; enter the necessary information in the primary template’s **Spoke Accounts** and **Spoke Regions** parameters.

Sample logs

This solution provides sample logs you can use for testing purposes. You can activate these sample logs from the AWS CloudFormation template, under the **Sample Logs** parameter. You can either activate this parameter during initial deployment, or post-deployment, when you update the stack. When activated,
the AWS CloudFormation template launches the `centralized-logging-demo` nested stack. For additional information about sample logs, refer to Sample logs (p. 21).

**Important**
Since the sample logs Apache web server is publicly accessible, we do not recommend deploying the demo AWS CloudFormation template in a production environment.

**Logging across accounts and Regions**

This solution creates Amazon CloudWatch Logs destinations that accepts log data from spoke AWS accounts and different AWS Regions. Customers can provide **Spoke Accounts** and additional AWS Regions (**Spoke Regions**) as template parameters during stack deployment. Setting up these parameters provides the necessary permission for the CloudWatch Logs destinations access policy to accept streaming log events across spokes accounts and different AWS Regions. Refer to the parameters table in Launch the stack (p. 13).

You can update these parameters, adding or removing **Spoke Accounts** or **Spoke Regions**, at any time post-deployment by updating the stack. Refer to Adding custom CloudWatch Logs (p. 23) for additional information about adding custom CloudWatch Logs.

**Note**
The latest version of this solution no longer requires you to install a separate spoke template in your secondary AWS accounts for multi-account usage. You can assign spoke accounts and spoke Regions from the primary template’s parameters; and run the command mentioned in the output section of the template from the spoke accounts or Regions.

**Solution updates**

If you are using a previous version of this solution (v3.x or earlier), we recommend deploying the latest version of the AWS CloudFormation template as a new stack. You can migrate your existing solution’s data to the new version by following the migration steps in Migrate your solution data (p. 30). After you have migrated your data, we recommend uninstalling the previous version to save on costs. Refer to Uninstall the solution (p. 31) for instructions to uninstall the primary AWS CloudFormation template.

**Regional deployments**

This solution uses Amazon Cognito which is available in specific AWS Regions only. You must launch this solution’s primary template in a Region that supports Amazon Cognito. (For the most current service availability by Region, refer to the AWS Regional Services List.)
AWS CloudFormation templates

This solution uses AWS CloudFormation to automate the deployment of the Centralized Logging solution in the AWS Cloud. It includes the following AWS CloudFormation templates, which you can download before deployment:

**centralized-logging-primary.template**: Use this template to launch the Centralized Logging solution and all associated components. The default configuration deploys an Amazon OpenSearch Service domain. The solution offers three deployment size options based on logging requirements, but you can also customize the template based on your specific needs.

**centralized-logging-demo.template**: Use this template to deploy in your spoke accounts or spoke Regions. This template deploys sample logs you can use for testing purposes. The default configuration deploys an Amazon Elastic Compute Cloud instance with a reference Apache server in an Amazon Virtual Private Cloud, an Amazon Simple Storage Service bucket, an AWS CloudTrail trail, and Amazon VPC flow logs.
Automated deployment

Before you launch the solution, review the architecture, configuration and other considerations in this guide. Follow the step-by-step instructions in this section to configure and deploy the solution into your account.

**Time to deploy:** Approximately 30 minutes

Deployment overview

Before deploying, review the available Deployment options (p. 10). Use the following steps to deploy this solution on AWS. For detailed instructions, follow the links for each step.

**Step 1. Launch the primary stack (p. 13)**

- Launch the AWS CloudFormation template into your AWS account.
- Enter values for required parameters: Admin Email Address, Spoke Accounts, Jumpbox Deployment, and Key Pair.
- Review the other template parameters, and adjust if necessary.

**Step 2. (Optional) Configure the Kibana dashboard (p. 16)**

- Import and configure the sample Kibana dashboard.

**Step 3. (Optional) Launch the demo stack in your spoke accounts or spoke Regions (p. 18)**

- Create sample resources that allow you to test the solution using a sample Kibana dashboard.

---

**Step 1. Launch the stack**

This automated AWS CloudFormation template deploys the Centralized Logging solution in the AWS Cloud.

**Note**

You are responsible for the cost of the AWS services used while running this solution. For more details, visit to the Cost (p. 2) section in this guide, and refer to the pricing webpage for each AWS service used in this solution.

1. Sign in to the AWS Management Console and use the button below to launch the centralized-logging-primary AWS CloudFormation template.

[Launch Solution]
Step 1. Launch the stack

Alternatively, you can download the template as a starting point for your own implementation.

2. The template launches 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 Create stack page, verify that the correct template URL is in the Amazon S3 URL text box and choose Next.

4. On the Specify stack details page, assign a name to your solution stack. For information about naming character limitations, refer to IAM and STS Limits in the AWS Identity and Access Management User Guide.

5. Under Parameters, review the parameters for this solution 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>OpenSearch Domain Name</td>
<td>centralized-logging</td>
<td>The name of the Amazon OpenSearch Service domain that this template creates. Note: Amazon OpenSearch Service domain names must start with a lowercase letter and be between three and 28 characters in length. Valid characters include a-z (lowercase only), 0-9, and the – (hyphen) symbol.</td>
</tr>
<tr>
<td>Cluster Size</td>
<td>Small</td>
<td>Select between three available Amazon OpenSearch Service cluster sizes: Small, Medium, and Large.</td>
</tr>
<tr>
<td>Admin Email Address</td>
<td>&lt;Requires input&gt;</td>
<td>Enter an email address for the Kibana dashboard administrator.</td>
</tr>
<tr>
<td>Spoke Accounts</td>
<td>&lt;Requires input&gt;</td>
<td>A comma delimited list of account IDs for log indexing. Enter the primary account ID along with any secondary account ID. Post-deployment, you can add or remove accounts by updating this parameter. For example, 111122223333, 444455556666</td>
</tr>
<tr>
<td>Spoke Regions</td>
<td>All</td>
<td>A comma delimited list of Regions for log indexing. Enter the AWS Regions from where you intend to collect logs. Post-deployment, you can add</td>
</tr>
</tbody>
</table>
Step 1. Launch the stack

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter or remove Regions by updating this parameter. For example, us-east-1, us-east-2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sample Logs</td>
<td>No</td>
<td>Choose whether to deploy the demo resources.</td>
</tr>
<tr>
<td>Jumpbox Deployment</td>
<td>No</td>
<td>Choose whether you want to deploy the Windows Jumpbox Server.</td>
</tr>
<tr>
<td>Key pair for jumpbox</td>
<td>&lt;Optional input&gt;</td>
<td>If you chose yes for Jumpbox Deployment, provide the existing key pair for Windows Jumpbox Server. Verify that you have access to the key pair.</td>
</tr>
<tr>
<td>WindowsAMI</td>
<td>/aws/service/ami-windows-latest/Windows_Server-2019-English-Full-Base</td>
<td>Windows AMI to be used for the jumpbox. This value does not need to be changed</td>
</tr>
</tbody>
</table>

6. Choose Next.
7. On the Configure stack options page, choose Next.
8. On the Review page, review and confirm the settings. Check the box acknowledging that the template will create AWS Identity and Access Management (IAM) resources.
9. Choose Create stack to deploy the stack.

You can view the status of the stack in the AWS CloudFormation Console in the Status column. You should receive a CREATE_COMPLETE status in approximately 30 minutes.

To view details for the stack resources, choose the Outputs tab. The following table describes the solution’s outputs.

<table>
<thead>
<tr>
<th>Key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Destination Subscription Command</td>
<td>This command can be run in the Spoke Accounts and Spoke Regions for streaming logs to the centralized account.</td>
</tr>
<tr>
<td>Kibana URL</td>
<td>The URL to access the visualizations from the Kibana dashboard.</td>
</tr>
</tbody>
</table>

**Note**
This solution deploys an AWS Lambda function, solution-helper, which runs only during initial configuration or when resources are updated or deleted. You can view the solution-helper function in the AWS Lambda console. Do not delete this function, as it is needed to manage associated resources.

Once the primary stack deployment completes, you will receive two emails:

- One email contains the temporary credentials for the Amazon Cognito user pool user created during deployment.
Step 2. (Optional) Configure the Kibana dashboard

If you activated the Sample Logs parameter during deployment of the AWS CloudFormation template, you can follow this step to configure the Kibana dashboard for the sample logs. Deploying the sample dashboard included with this solution creates visualizations that combine data from the Amazon VPC flow logs, the Apache web server, and AWS CloudTrail to create a centralized view of all relevant logging data.

After the Centralized Logging solution is deployed, you will receive a verification email with a user name and password to access the Kibana dashboard. Use the following steps to set up a Windows Jumpbox Server, sign in to Kibana, and import the solution's sample dashboard.

Before you can begin, you must launch a Windows Jumpbox Server inside the public subnet of the Amazon VPC created by the solution. For information about accessing your Windows Jumpbox Server, refer to Accessing Jumpbox (p. 22).

1. Use remote desktop protocol (RDP) to sign in to the Windows Jumpbox Server.
2. Sign in to the AWS CloudFormation console and select this solution's primary stack.
3. Select the Outputs tab and copy the KibanaURL link for the Kibana dashboard.
4. In your Windows Jumpbox Server, enter the KibanaURL in a web browser. We recommend using either Mozilla Firefox or Google Chrome.
5. Sign in to the Kibana dashboard with the username and password from your verification email. Note that you will be prompted to change the password when you sign in for the first time.

Import dashboard

1. Download the dashboard configuration file (dashboard.ndjson) from this solution’s Amazon S3 bucket. You will use this file to configure your first dashboard.
2. On the Kibana home page, choose Connect to your Elasticsearch index.
3. Choose Saved Objects from the left menu pane.
4. From the Saved Objects page, choose Import and select the dashboard.ndjson file. Choose Done after importing the file.
5. Under the Saved Objects section, a resulting list of the index and visualizations will appear (as shown in Figure 6.). Choose Basic to view this solution’s sample dashboard. Test the following features on the dashboard:

**Note**
If the dashboard does not show up and Kibana prompts you to create index, do not create an index. Go back to the Saved objects page and choose Basic dashboard. The index might take a couple of minutes to propagate.

6. Test the following features on the dashboard:

- Adjust the data time period by choosing the clock icon in the upper-right corner of the screen.
- Adjust the interval for the data refresh rate by choosing the Auto-refresh option.
Centralized Logging Implementation Guide

Step 3. (Optional) Launch the demo stack in your spoke accounts or spoke Regions

This solution provides a demo template that you can launch in your spoke accounts.

1. In your spoke account, or from a spoke Region, sign in to the AWS Management Console and use the button below to launch the centralized-logging-demo AWS CloudFormation template.

   ![Launch Solution Button]

Alternatively, you can download the template as a starting point for your own implementation.

2. The template launches 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 Create stack page, verify that the correct template URL is in the Amazon S3 URL text box and choose Next.

4. On the Specify stack details page, assign a name to your solution stack. For information about naming character limitations, refer to IAM and STS Limits in the AWS Identity and Access Management User Guide.
5. Under **Parameters**, review the parameter for this demo template and update as necessary.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CloudWatch Logs Destination ARN</td>
<td>&lt;Requires input&gt;</td>
<td>Enter the ARN for the CloudWatch Logs destination that was created in the primary account. This value can be identified in the primary template’s <strong>Output</strong> tab from the <strong>Destination Subscription Command</strong> output.</td>
</tr>
</tbody>
</table>

6. Choose **Next**.

7. On the **Options** page, choose **Next**.

8. On the **Review** page, review and confirm the settings. 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 receive a **CREATE_COMPLETE** status in approximately five minutes.
Additional resources

<table>
<thead>
<tr>
<th>AWS services</th>
<th>Additional documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>• AWS CloudFormation</td>
<td>• Kibana User Guide</td>
</tr>
<tr>
<td>• Amazon OpenSearch Service</td>
<td></td>
</tr>
<tr>
<td>• AWS Lambda</td>
<td></td>
</tr>
<tr>
<td>• Amazon S3</td>
<td></td>
</tr>
<tr>
<td>• Amazon Kinesis Data Streams</td>
<td></td>
</tr>
<tr>
<td>• Amazon Cognito</td>
<td></td>
</tr>
<tr>
<td>• Amazon CloudWatch</td>
<td></td>
</tr>
<tr>
<td>• Amazon Elastic Compute Cloud</td>
<td></td>
</tr>
<tr>
<td>• Amazon Kinesis Data Firehose</td>
<td></td>
</tr>
<tr>
<td>• AWS Identity and Access Management</td>
<td></td>
</tr>
</tbody>
</table>
Sample logs

The Centralized Logging solution includes a `centralized-logging-demo` AWS CloudFormation template that deploys sample logs you can use for testing purposes. This demo template launches an Amazon Elastic Compute Cloud (Amazon EC2) instance with a reference Apache server that hosts a simple web application in an Amazon Virtual Private Cloud (Amazon VPC). During initial launch, the Amazon CloudWatch Logs agent is automatically installed on the instance, which is used to direct raw log data to Amazon CloudWatch.

VPC Flow Logs are activated in the VPC to capture information about IP traffic to, from, and within the network. Customers can use this example to activate VPC Flow Logs in other VPCs; this data is automatically published to a log group in Amazon CloudWatch Logs.

The demo template activates AWS CloudTrail and creates a trail for the account. It also creates an Amazon Simple Storage Service (Amazon S3) bucket to store CloudTrail logs, which are automatically delivered to Amazon CloudWatch.

The demo template also configures the sample log groups with CloudWatch subscription filters to stream log events to the centralized account.

This demo template can be automatically deployed when you activate the Sample Logs parameters in the primary solution template.

Sample logs Apache web server

Note that the sample logs Apache web server this solution deploys is publicly accessible on port 80. If you modify this sample logs web server for production use, we recommend that you use HTTPS by enabling Transport Layer Security (TLS) and add authentication.

**Important**

Since the sample logs Apache web server is publicly accessible, we do not recommend deploying the demo AWS CloudFormation template in a production environment.
Accessing Jumpbox

This solution launches an Amazon OpenSearch Service domain inside an Amazon VPC. To access and view the Kibana dashboard, a Windows Jumpbox Server (bastion host) in a public subnet must be set up in the same VPC as the Amazon OpenSearch Service domain.

This solution can automatically deploy a Windows Jumpbox Server if you set the Jumpbox Deploy parameter to Yes when you launch the stack (p. 13). Optionally, you can launch your own Windows Jumpbox Server. Once your Windows Jumpbox Server is running, you can take the following steps to connect to it using your remote desktop protocol (RDP):

1. Sign in to the Amazon Elastic Compute Cloud console.
2. From the Amazon EC2 dashboard, choose the CL-PrimaryStack/CL-Jumpbox/JumpboxEC2 instance.
3. Select the Security tab and click on the referenced security group.
4. From the Inbound rule tab, choose Edit inbound rules.
5. In the Edit inbound rules page, choose Add rule. A new inbound rule type displays. Complete the following actions:
   • Under the Type column, select RDP from the drop-down menu.
   • Under the Source column, select My IP from the drop-down menu. This option allows you to access the Windows Jumpbox Server.
   • Choose Save rules.
Adding custom CloudWatch Logs

The Centralized Logging solution allows you to add custom CloudWatch Logs groups to the solution's Amazon OpenSearch Service domain. Use the following procedure to add your log groups.

**Note**
Before taking the following steps, ensure that you have added the account and the AWS Region to the primary stack's Spoke Accounts and Spoke Regions parameter. The primary account requires the appropriate permissions to receive logs from the secondary accounts and Regions.

The following steps must be taken in a spoke account or in a spoke Region.

1. Sign in to the Amazon CloudWatch console.
2. From the left menu pane, under Logs, select Log groups.
3. Identify the applicable log group and take a note of the name of the log group.
4. Open an AWS Command Line Interface (AWS CLI) session.
5. Run the following `put-subscription-filter` command, replacing the following placeholders:

   ```bash
   
   - `<REGION>`: Region of your CloudWatch Logs group
   - `<PRIMARY_ACCOUNT_ID>`: account ID of the centralized account
   - `<MY_LOG_GROUP>`: name of the CloudWatch Logs group (as noted in step 3)
   - `<MY_FILTER>`: name for the filter pattern
   - `<MY_PATTERN>`: pattern for the subscription filter
   - `<MY_PROFILE>`: name of an AWS CLI profile that contains appropriate credentials for creating subscription filter in your preferred Region

The logs should will stream to the centralized account.
Adding social identity providers

To activate Google sign-in for your web application, you can create a Google Developers console project for your application.

1. Access the Google Developers console and create a new project.
2. Under APIs & Services, OAuth consent screen, create a custom message for users who use your app to request access to their private data.
   - Enter the name of your app.
   - Under App domain, enter the VPC endpoint URL in the application’s home page.
   - Under Authorized domains, enter amazonaws.com and amazoncognito.com.
   - Choose Save then Continue to complete the setup.
3. From the left menu pane, choose Credentials.
4. Create your OAuth 2.0 credentials by choosing OAuth client ID from the Create credentials dropdown list.
   - From the Application type drop-down list, choose Web application and create a name for your application.
   - In the Authorized JavaScript origins field, enter your user pool domain.
     ```
     https://<your-user-pool-domain>
     ```
   - In the Authorized Redirect URIs, enter your user pool domain with the /oauth2/idpresponse endpoint.
     ```
     https://<your-user-pool-domain>/oauth2/idpresponse
     ```
5. Choose Create.
6. Record the OAuth client ID and client secret for use in the next steps.
7. Choose OK.

Take the following steps to configure an Amazon Cognito user pool social identity provider:

1. Sign in to the Amazon Cognito console and choose Manage User Pools.
2. Choose the user pool created by the solution.
3. On the left menu bar, choose Identity providers and choose a social identity provider, for example, Google.
4. Enter the OAuth client ID and client secret.
5. Enter the names of the scopes that you want to authorize, for example, email and choose Enable.
6. Choose App client settings from the navigation bar and select Google as one of the Enabled Identity Providers for your user pool app. Choose Save changes.
7. On the Attribute mapping tab, add mappings for email, as follows:
   a. Select the check box to choose the Google attribute name, which is an email address.
   b. Choose the destination user pool attribute from the drop-down list, which is listed as an email address.
   c. Choose Save changes.
   d. Choose Go to summary.
To add more social identity providers, refer to Adding Social Identity Providers to a User Pool in the Amazon Cognito Developer Guide.
Troubleshooting

The Centralized Logging solution logs error, warning, informational, and debugging messages for the solution’s AWS Lambda functions. To choose the type of messages to log, find the applicable function in the Lambda console and change the `LOG_LEVEL` environment variable to the applicable type of message.

<table>
<thead>
<tr>
<th>Level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERROR</td>
<td>Logs will include information on anything that causes an operation to fail.</td>
</tr>
<tr>
<td>WARNING</td>
<td>Logs will include information on anything that can potentially cause inconsistencies in the function but might not necessarily cause the operation to fail. Logs will also include ERROR messages.</td>
</tr>
<tr>
<td>INFO</td>
<td>Logs will include high-level information about how the function is operating. Logs will also include ERROR and WARN messages.</td>
</tr>
<tr>
<td>DEBUG</td>
<td>Logs will include information that might be helpful when debugging a problem with the function. Logs will also include ERROR, WARNING, and INFO messages.</td>
</tr>
</tbody>
</table>

Common Errors

Validation error with the AWS CloudFormation primary template

Error

A **Template validation error** occurs before you deploy the primary stack.

![Error](image)

Figure 8: Template validation error

Resolution

Verify that you are deploying the stack in an AWS Region that supports Amazon Cognito.
1. In the primary account, navigate to the AWS Management Console.
2. In the console navigation bar, use the Region selector to choose an AWS Region that supports Amazon Cognito.

   **Note**
   For the most current service availability by region, refer to the AWS Regional Services List.
3. Follow procedure to Launch the stack (p. 13).

**Error in creating a CloudWatch Logs subscription filter**

**Error**

An AccessDeniedException error occurs when calling the PutSubscriptionFilter operation:

```
```

**Resolution**

This error is due to an access policy inappropriately set on the CloudWatch Logs destination. Take the following steps to update the access policy:

1. Review the access policy for CloudWatch Logs destination in the primary account and verify that it lists the spoke account as the principal by running the following command:

   ```bash
   aws logs describe-destinations -region <REGION>
   ```

2. If the spoke account is not listed as the principal, update the primary stack Spoke Account parameter with the secondary account id.

**CloudWatch Logs group already exists**

**Error**

A CREATE_FAILED error occurs in Amazon Kinesis Data Firehose CloudWatch Logs group.

![Figure 9: Kinesis Data Firehose error](image)

**Resolution**

This issue occurs when the Amazon CloudWatch Logs with same identifier already exists in the account. To resolve this error, either delete the existing CW Log Group or change the name of the erroneous log group in the template and then redeploy the stack.
Invalid Region parameter entered in the primary template

Error

A CREATE_FAILED stack error occurs in the CloudWatch Logs:

```
Failed to create resource. failed to put cw logs destinations, please see in cw logs for more details
Logs: /aws/lambda/cl-sing05-HelpelLambdaCA947A74-S9GNqM5VLISVRs at sendResponse
(\var/task/index.js:332:15) at Runtime.exports.handler
(\var/task/index.js:147:18) at processTicksAndRejections
(internal/process/task_queues.js:97:5)
```

Figure 10: CloudWatch Logs error

CloudWatch Logs error:

```
[ERROR] [helper/areRegionsValid] invalid region provided
[ERROR] [helper/putDestination] Error: invalid regions
```

Resolution

This issue occurs when the provided value for a Region parameter is not valid. To resolve this issue, update the Spoke Regions parameter in the primary stack with the correct Region names (us-east-1, eu-west-2, etc.). Run the following CLI command to get a list of Regions that are used in your account.

```
aws ec2 describe-regions --query Regions[*].RegionName --profile <MY_PROFILE>
```

An Amazon VPC limit error

Error

A CREATE_FAILED stack error occurs when attempting to create an Amazon VPC:

```
ESVPC3CED2A7 CREATE_FAILED
```

Figure 11: Amazon VPC limit error

Resolution

This solution creates a new Amazon VPC containing an Amazon OpenSearch Service domain. This issue occurs when you reach your VPC limit per Region in your AWS account. Either delete an existing VPC or increase the VPC limit in the target Region.
An AWS CloudFormation nested stack stays in a REVIEW_IN_PROGRESS status

Error

When you activate the primary template's Sample Logs parameter, this solution deploys a nested stack containing demo resources and another nested stack with change set. This change set nested stack may remain in a REVIEW_IN_PROGRESS status.

**Resolution**

Delete the change set nested stack. Ensure that the nested stack with demo resources is in CREATE_COMPLETE state.
Migrate your solution data

If you use an earlier version (v3.x and earlier) of the Centralized Logging solution, and you want to move to the new version of the solution, you must migrate data from your existing Amazon OpenSearch Service domain to the newly provisioned OpenSearch Service domain to avoid losing data from your existing domain.

**Important**

Do not update the solution stack.

Take the following steps to migrate your data:

1. Deploy the new version of the primary AWS CloudFormation template in the same account and AWS Region as your current primary stack.
2. Use a snapshot to migrate the indexed data on your Amazon OpenSearch Service domain.
3. After the migration is complete, delete the old AWS CloudFormation stack to stop incurring charges.
Uninstall the solution

You can uninstall the Centralized Logging solution using either the AWS Management Console or the AWS Command Line Interface.

**Warning**
Uninstalling the solution does not delete the Amazon OpenSearch Service domain provisioned by the solution. To stop incurring cost on the domain, the domain must be deleted as per user discretion, when it is safe to delete it. We recommend taking a snapshot on the Amazon OpenSearch Service domain.

Amazon VPC, Amazon OpenSearch Service domain, Amazon Simple Storage Service (Amazon S3) buckets, and Amazon CloudWatch Logs created by this solution are retained. These resources must be deleted manually. Specifically, the following resources are retained:

- Amazon VPC containing the Amazon OpenSearch Service domain
- Amazon OpenSearch Service domain
- Access Logs S3 bucket
- CloudWatch Logs S3 bucket, which contains the indexed data from Amazon OpenSearch Service
- Kinesis Data Firehose Logs group: CloudWatch Logs group with Kinesis Data Firehose logs
  - Kinesis Data Firehose Amazon OpenSearch Service log stream: CloudWatch Logs stream with Amazon OpenSearch Service delivery logs
  - Kinesis Data Firehose S3 log stream: CloudWatch Logs stream with Amazon OpenSearch Service delivery logs
- Amazon VPC flow log group: Amazon VPC flow logs for the OpenSearch Service domain VPC

### Using the AWS Management Console

1. Sign in to the AWS CloudFormation console.
2. Select this solution's installation stack.
3. Choose Delete.

### Deleting the Amazon S3 buckets

This solution is configured to retain the Amazon S3 buckets if you decide to delete the AWS CloudFormation stack to prevent accidental data loss. After uninstalling the solution, you can manually delete the S3 buckets if you do not need to retain the data. Use the following steps to delete the Amazon S3 buckets.

1. Sign in to the Amazon S3 console.
2. Choose Buckets from the left navigation pane.
3. Locate the `<stack-name>` S3 buckets.
4. Select one of the S3 buckets and choose Delete.

Repeat the steps until you have deleted all the `<stack-name>` S3 buckets.
Alternatively, you can configure the AWS CloudFormation template to delete the Amazon S3 buckets automatically. Before deleting the stack, change the deletion behavior in the AWS CloudFormation DeletionPolicy attribute.

Deleting the CloudWatch Logs

This solution retains the CloudWatch Logs if you decide to delete the AWS CloudFormation stack to prevent against accidental data loss. After uninstalling the solution, you can manually delete the logs if you do not need to retain the data. Use the following procedure to delete the CloudWatch Logs.

1. Sign in to the Amazon CloudWatch console.
2. Choose Log Groups from the left navigation pane.
3. Locate the log groups created by the solution.
4. Select one of the log groups.
5. Choose Actions and then choose Delete.

Repeat the steps until you have deleted all the solution log groups.

Deleting the Amazon OpenSearch Service domain

Use the following procedure to delete the Amazon OpenSearch Service domain from the console.

1. Sign in to the Amazon OpenSearch Service console.
2. Under Domains, select the movies domain.
3. Choose Delete and confirm deletion.

Deleting the Amazon VPC

Use the following procedure to delete the Amazon VPC from the console.

1. Sign in to the Amazon VPC console.
2. In the navigation pane, choose the solution created VPCs (CL-PrimaryStack/ESVPC)
3. Select the VPC, choose Actions, and then choose Delete VPC.

Using AWS Command Line Interface

Determine whether the AWS Command Line Interface (AWS CLI) is available in your environment. For installation instructions, refer to What Is the AWS Command Line Interface in the AWS CLI User Guide. After confirming that the AWS CLI is available, run the following command.

```
$ aws cloudformation delete-stack --stack-name <installation-stack-name>
```
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 activated, the following information is collected and sent to AWS:

- **Solution ID**: The AWS solution identifier
- **Unique ID (UUID)**: Randomly generated, unique identifier for each Centralized Logging solution deployment
- **Timestamp**: Data-collection timestamp
- **Cluster Size**: Size of the Amazon OpenSearch Service cluster the solution will deploy
- **Total Item Size**: The total size (in bytes) of the items put on Kinesis Data Firehose

AWS owns the data gathered through this survey. Data collection is subject to the AWS Privacy Policy. To opt out of this feature, modify the AWS CloudFormation template mapping section from:

```json
"Mappings": {  
  "CLMap": {  
    "Metric": {  
      "SendAnonymousMetric": "Yes"  
    }  
  }  
}
```

to:

```json
"Mappings": {  
  "CLMap": {  
    "Metric": {  
      "SendAnonymousMetric": "No"  
    }  
  }  
}
```
Source code

Visit our GitHub repository to download the templates and scripts for this solution, and to share your customizations with others.
## Revisions

<table>
<thead>
<tr>
<th>Date</th>
<th>Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>November 2016</td>
<td>Initial release.</td>
</tr>
<tr>
<td>February 2018</td>
<td>Added cross-account and cross-region functionality and upgraded Amazon OpenSearch Service version to 6.0.</td>
</tr>
<tr>
<td>November 2018</td>
<td>Added Amazon Cognito for Kibana dashboard user authentication, changed custom sizing of the Elasticsearch cluster, and added information on troubleshooting.</td>
</tr>
<tr>
<td>December 2019</td>
<td>Added information on support for Node.js update.</td>
</tr>
<tr>
<td>December 2020</td>
<td>Updated the solution architecture with Kinesis Data Streams, Kinesis Data Firehose, and CloudWatch Logs destination; added support for Elasticsearch version 7.7; information added in social identity provider and troubleshooting sections; for more information about changes for v4.0.0, refer to the CHANGELOG.md file in the GitHub repository.</td>
</tr>
<tr>
<td>June 2021</td>
<td>Document enhancements, including updates to diagrams and architecture overview section.</td>
</tr>
<tr>
<td>September 2021</td>
<td>Updated all instances of Amazon ElasticSearch Service to Amazon OpenSearch Service.</td>
</tr>
<tr>
<td>December 2021</td>
<td>Updated Node.js version to &gt;=12, added support for + in admin email address, added support for multiple deployments, refactored code to improve code-quality score. For more information about changes for v4.0.1, refer to the CHANGELOG.md file in the GitHub repository.</td>
</tr>
</tbody>
</table>

Contributors

- Garvit Singh
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 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 products or services are provided “as is” without warranties, representations, or conditions of any kind, whether express or implied. AWS 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.

Centralized Logging is licensed under the terms of the of the Apache License Version 2.0 available at The Apache Software Foundation.