AWS Limit Monitor

Implementation Guide

AWS Limit Monitor: Implementation Guide

Table of Contents

AWS Limit Monitor

AWS Implementation Guide

AWS Solutions Builder Team

September 2016 (last update (p. 23): October 2020)

This implementation guide discusses architectural considerations and configuration steps for deploying the Amazon Web Services (AWS) Limit Monitor solution in the 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 in the AWS Cloud.

Overview

To help provide highly-available, reliable, and robust services to all of our customers, as well as minimize billing risk for new customers, Amazon Web Services (AWS) maintains service limits for each account. Nearly every AWS service is regulated in terms of how many resources you can launch within a specific AWS Region at a given time. However, there are times when even the most experienced AWS customers can unexpectedly hit service limits.

To help customers more actively track their AWS resource usage against service limits, AWS offers the AWS Limit Monitor solution, a reference implementation that automatically provisions the services necessary to proactively track resource usage and send notifications as you approach limits. The solution is easy-to-deploy and leverages AWS Trusted Advisor Service Limits checks and Service Quotas to help you display your usage and limits for specific AWS services, and centrally manage your limits. With the AWS Limit Monitor, you can receive email notifications or notifications can be sent to your existing Slack channel, enabling you to request limit increases or shut down resources before the limit is reached.

You can also use this solution to automatically monitor Amazon Elastic Compute Cloud (Amazon EC2) virtual central processing unit-based (vCPU-based) On-Demand Instance limits. For more information, see Supported Service Limit Checks (p. 5).

> **Note**
> To use this solution, each account must have a Business- or Enterprise-level AWS Support plan in order to gain access to the Trusted Advisor Service Limits checks.

Cost

You are responsible for the cost of the AWS services used while running the AWS Limit Monitor. The total cost for running this solution depends on the interval at which you run the AWS Lambda functions, and the number of accounts you monitor, and whether you use this solution to monitor vCPU-based instance limits.

If you do not use this solution to monitor vCPU limits, the cost for running this solution in your primary account in US East (N. Virginia) is approximately **$2.36 per month**, and **$0.89 per month per secondary account** as of the date of publication.

If you use this solution to monitor vCPU limits, the cost for running this solution in your primary account in US East (N. Virginia) is approximately **$9.50 per month**, and **$8.03 per month per secondary account** as of the date of publication.

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

Architecture Overview

Deploying this solution builds the following environment in the AWS Cloud.

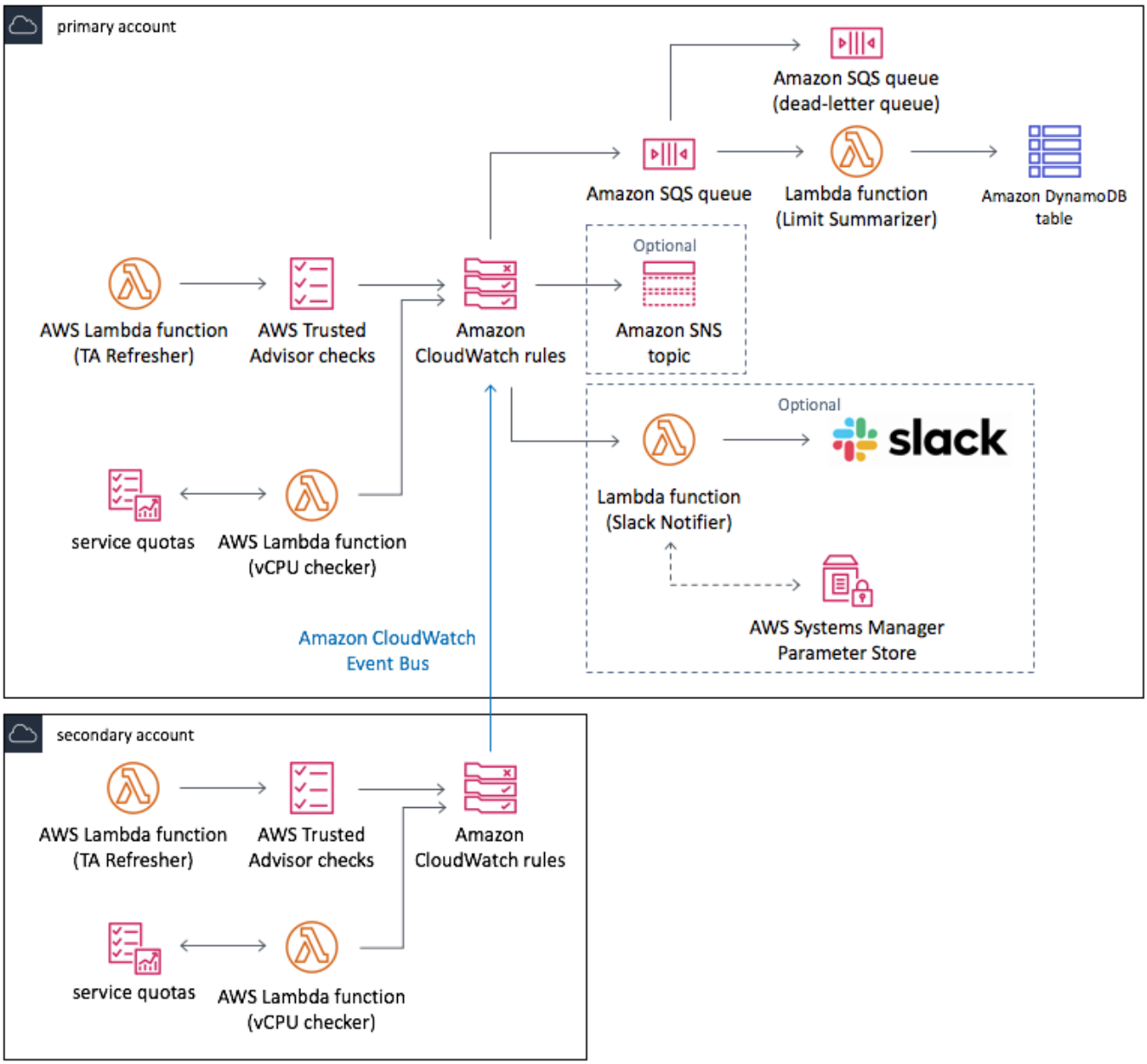

Figure 1: AWS Limit Monitor architecture

The AWS Limit Monitor includes a template that you deploy in your primary account. This template launches an AWS Lambda function that runs once every 24 hours. (You can modify the AWS CloudFormation template to change how often the refresh Lambda function is invoked. For more information, see Appendix B (p. 19).) The Lambda function refreshes the AWS Trusted Advisor Service Limits checks to retrieve the most current utilization and limit data through API calls. Trusted Advisor calculates usage against the limit to determine whether the status is OK (less than 80% utilization), WARN (between 80% and 99% utilization), or ERROR (100% utilization).

If you opt in to monitor Amazon Elastic Compute Cloud (Amazon EC2) virtual central processing unit-based (vCPU-based) limits, the template launches another Lambda function that runs every five minutes. The function checks Service Quotas to retrieve vCPU usage and limit data for every AWS Region. The

function calculates vCPU usage against limits to determine whether the status is OK (less than 80% utilization), WARN (between 80% and 99% utilization), or ERROR (100% utilization).

Amazon CloudWatch Events captures the status events from Trusted Advisor and the vCPU monitoring Lambda function, and uses a set of CloudWatch Events rules to send the status events to all the targets you choose during initial deployment of the solution: an Amazon Simple Queue Service (Amazon SQS) queue, an Amazon Simple Notification Service (Amazon SNS) topic (optional), or a Lambda function for Slack notifications (optional).

If you enable Slack notifications during initial deployment, the solution will launch a Lambda function that sends notifications to your existing Slack channel. An AWS Systems Manager Parameter Store will also be deployed to provide highly available, secure, durable storage for your Slack WebHook URL which is used to send messages to the Slack channel. For more information, see Slack Integration (p. 5).

Amazon SQS receives all the OK, WARN, and ERROR status events and sends them to an Amazon DynamoDB table that stores the events. By default, Amazon SNS and Slack receive only WARN and ERROR status events. But, you can customize the notifications for your specific needs.

The solution also includes a secondary template you can deploy in secondary accounts. This template launches a Lambda function that refreshes the Trusted Advisor Service Limits check in the secondary account. If enabled, this template also launches a Lambda function to check Service Quotas for vCPU limits. CloudWatch Events in the secondary account captures the status events from both functions and sends those events to the primary account using the CloudWatch Event Bus. Once those events are received in the primary account, the CloudWatch Events rules send the events to your chosen targets.

Note
AWS CloudFormation resources are created from AWS Cloud Development Kit (AWS CDK) components.

Considerations

Supported Service Limit Checks

The AWS Limit Monitor leverages AWS Trusted Advisor to check usage against service limits. For a list of service limits that Trusted Advisor checks, see the Trusted Advisor FAQs. For a list of the default service limits and how to request a limit increase, see AWS Service Limits.

Amazon Elastic Compute Cloud (Amazon EC2) is transitioning On-Demand Instance limits from the current instance count-based limits to the new virtual central processing unit-based (vCPU-based) On-Demand Instance limits to simplify the limit management experience for AWS customers. Beginning September 24, 2019, you can opt in to vCPU-based instance limits. Amazon EC2 will be migrating instance limits to vCPUs starting October 24, 2019, and current count-based instance limits will not be available or supported after November 8, 2019. For more information, see EC2 On-Demand Instance Limits. The Limit Monitor can also monitor vCPU-based On-Demand Instance limits.

Until October 24, 2019, you can choose whether to enable vCPU-limit monitoring for this solution. After October 24, 2019, this solution will automatically monitor vCPU limits.

Customization

By default, this solution checks all service limits that Trusted Advisor checks. You can modify the AWS CloudFormation template mappings to specify which service limits you want to check in both the primary and secondary accounts. For more information, see Change the Monitored Services (p. 19).

Notifications

You can specify whether you want to receive notifications. If you choose to receive notifications, you can choose whether you want to be notified for WARN, ERROR, or both status events. If you choose not to receive notifications, the solution still stores usage information in Amazon DynamoDB.

Slack Integration

This solution includes an optional configuration to send notifications to your existing Slack channel. To use this feature, you must have an existing Slack channel, and you must specify parameters for the Slack webhook and channel. These parameters are stored in the AWS Systems Manager Parameter Store, which provides secure, hierarchical storage for configuration data management and secrets management. If you specify parameters that exist in the Parameter Store, the solution will use the existing parameters. It will not create new ones.

If you specify parameters that don't already exist in Parameter Store, the solution will create the parameters with dummy values. Then, you must add your unique Slack webhook URL to the solutions the Parameter Store to receive Slack notifications. For more information, see Step 3 (p. 13).

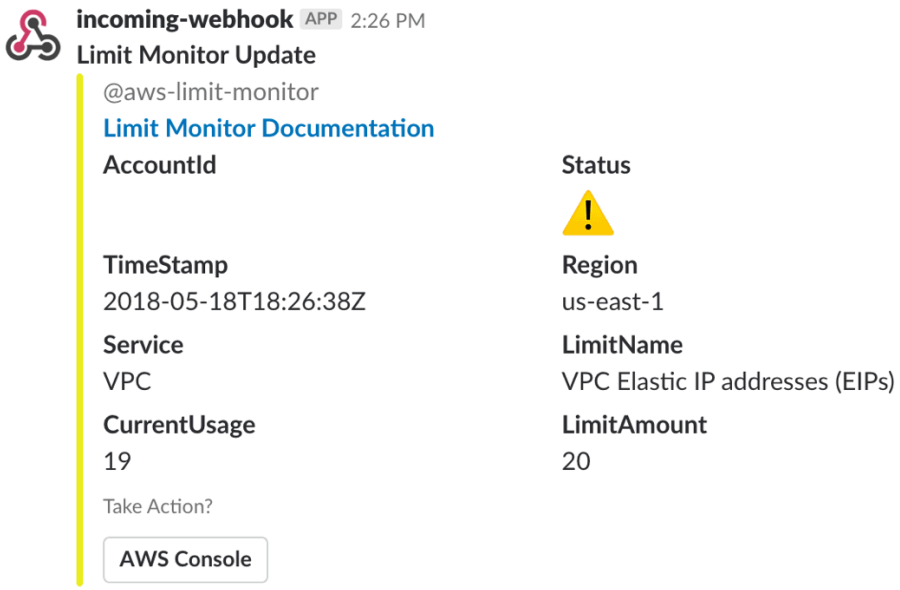

Figure 2: Sample Slack Notification

Amazon SQS Dead-Letter Queue

The Limit Monitor solution also deploys an Amazon Simple Queue Service (Amazon SQS) dead-letter queue. The `Limit Summarizer` AWS Lambda function attempts to process messages three times. If it cannot process the message after three attempts, the message is sent to the dead-letter queue where you can debug.

Solution Updates

Update the main stack

Take the following steps to update your main AWS CloudFormation stack to the current version.

1. From your main account where the primary AWS Limit Monitor template is deployed, sign in to the AWS CloudFormation console.
2. From the **Stacks** page, select your primary AWS Limit Monitor stack and choose **Update**.
3. On the **Update stack** page, verify that **Replace current template** is selected.
   - In the **Specify template** section, select **Amazon S3 URL**.
   - Copy the link of the latest template.
   - Paste the link in the **Amazon S3 URL** box.
   - Verify that the correct template URL shows in the **Amazon S3 URL** text box and choose **Next**.
4. On the **Specify stack details** page, under **Parameters**, review the parameters for the template and modify them as necessary. Refer to Step 1. Launch the Stack (p. 9) for details about the parameters.
5. Choose **Next**.

6. On the **Configure stack options** page, choose **Next**.
7. 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.
8. Choose **View change set** and verify the changes.
9. Choose **Update stack** 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 **UPDATE_COMPLETE** in approximately five minutes.

Update the spoke stack

Take the following steps to update the spoke AWS CloudFormation stack in your secondary accounts to the current version.

1. From your secondary account where the AWS Limit Monitor spoke template is deployed, sign in to the AWS CloudFormation console.
2. From the **Stacks** page, select your primary AWS Limit Monitor stack and choose **Update**.
3. On the **Update stack** page, verify that **Replace current template** is selected.
   - In the **Specify template** section, select **Amazon S3 URL**.
   - Copy the link of the latest template.
   - Paste the link in the **Amazon S3 URL** box.
   - Verify that the correct template URL shows in the **Amazon S3 URL** text box and choose **Next**.
4. On the **Specify stack details** page, under **Parameters**, review the parameters for the template and modify them as necessary. Refer to Step 1. Launch the Stack (p. 9) for details about the parameters.
5. Choose **Next**.
6. On the **Configure stack options** page, choose **Next**.
7. 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.
8. Choose **View change set** and verify the changes.
9. Choose **Update stack** to deploy the stack.

Repeat these steps to update additional secondary accounts containing the spoke template.

About Node.js versions

AWS Limit Monitor version 5.3.0 and earlier versions use the Node.js 8.10 runtime, which reached end-of-life on December 31, 2019. AWS Lambda now blocks both the create operation and the update operation. For more information, see Runtime Support Policy in the *AWS Lambda Developer Guide*. To continue using this solution with the latest features and improvements, you must update the stack (p. 6).

Regional Deployments

You must launch this solution's AWS CloudFormation templates in the US East (N. Virginia) Region. However, once deployed, the solution will monitor limits for all regions in the account.

AWS CloudFormation Templates

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

limit-monitor.template: Use this template to launch the AWS Limit Monitor and all associated components. The default configuration deploys AWS Lambda functions, Amazon CloudWatch Events rules, Amazon Simple Queue Service queues, and an Amazon DynamoDB table. If you enable notifications, an Amazon Simple Notification Service topic will be deployed. If you enable Slack notifications, the template also deploys an additional Lambda function and an AWS Systems Manager Parameter Store. You can also customize the template based on your specific needs. Refer to the README.md file in the GitHub repository for guidance to customize the template.

limit-monitor-spoke.template: Use this template to launch the AWS Limit Monitor and all associated components in secondary accounts. The default configuration deploys Lambda functions and CloudWatch Events rules, but you can customize the template based on your specific needs. Refer to the README.md file in the GitHub repository for guidance to customize the template.

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 instructions in this section to configure and deploy the AWS Limit Monitor into your account.

Time to deploy: Approximately five minutes

Prerequisites

- To use this solution, each account must have a Business- or Enterprise-level AWS Support plan in order to gain access to the Trusted Advisor Service Limits checks.
- To use this solution's Slack notification functionality, you must have an existing Slack channel.

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 (p. 9)

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

Step 2. Launch the Spoke Stack (Optional) (p. 12)

- Launch the AWS CloudFormation template into secondary AWS accounts
- Review the other template parameters and adjust if necessary

Step 3. Configure Slack Notifications (Optional) (p. 13)

- Add the webhook URL to the AWS Systems Manager Parameter Store

Step 1. Launch the Stack

This automated AWS CloudFormation template deploys AWS Limit Monitor into your account. Ensure that your account has a Business- or Enterprise-level AWS Support plan, and that you have an existing Slack channel, if necessary, before launching the stack.

> **Note**
> You are responsible for the cost of the AWS services used while running this solution. See the Cost (p. 2) section for more details. For full details, see the pricing webpage for each AWS service you will be using in this solution.

1. Sign in to the AWS Management Console and click the button below to launch the `limit-monitor` AWS CloudFormation template.

You can also download the template as a starting point for your own implementation. Refer to the README.md file in the GitHub repository for guidance to customize the template.

2. The template is launched in the US East (N. Virginia) Region by default.

   Note
   You must launch this solution in the US East (N. Virginia) Region.

3. On the **Create stack** page, verify that the correct template URL shows in the **Amazon S3 URL** text box and choose **Next**.
4. On the **Specify stack details** page, assign a name to your solution stack.
5. Under **Parameters**, review the parameters for the template, and modify them as necessary.

Parameter	Default	Description
Account List	<Optional Input>	List of account IDs for limit monitoring. Note that the format is double quotation marks and comma separated (for multiple values), and the value must match the regular expression: `^"\d{12}"(,"\d{12}")*$\|(^\s*)$`. Enter the secondary account IDs in this parameter before you deploy the spoke template in secondary accounts. To add accounts after you launch the primary template, update the **Account List** parameter in the primary stack with the secondary account IDs. Then, update the primary stack and deploy the spoke template in the secondary accounts. **Note** If you leave this parameter blank, the solution will only monitor limits in the primary account. If you enter a secondary account ID, you must also enter the primary account ID in this parameter.
Email Notification Level	*"WARN", "ERROR"*	Choose the status event level(s) that will trigger notifications. For example,

Parameter	Default	Description
		`"WARN", "ERROR"`. Note that the format is double quotation marks and comma separated (for more than one value). **Note** Leave this parameter blank if you do not want to receive Amazon Simple Notification Service (Amazon SNS) notifications. Note that the SNS notification components will not be deployed.
Email Address	<Optional Input>	A valid email address to receive Amazon SNS notifications for service limit alerts.
Slack Notification Level	*"WARN", "ERROR"*	Choose the status event level(s) that will trigger Slack notifications. For example, `"WARN", "ERROR"`. Note that the format is double quotation marks and comma separated (for multiple values). **Note** Leave this parameter blank if you do not want to receive Slack notifications. Note that the Slack notification components will not be deployed.
Slack Hook URL Key Name	<Optional Input>	The AWS Systems Manager parameter key for the incoming Slack webhook. **Note** If the parameter key does not exist in the parameter store, the solution will create one with a dummy value. The parameter name cannot begin with either `aws` or `ssm` prefixes (case-insensitive).

Parameter	Default	Description
Slack Channel Key Name	<Optional Input>	The AWS Systems Manager parameter key for the Slack channel **Note** If the parameter key does not exist in the parameter store, the solution will create one with a dummy value. The parameter name cannot begin with either `aws` or `ssm` prefixes (case-insensitive).

6. Choose **Next**.
7. On the **Configure stack 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 stack** to deploy the stack.

   You can view the status of the stack in the AWS CloudFormation console in the Status column. You should see a status of **CREATE_COMPLETE** in approximately five minutes.
10. In the subscription notification email, select the **SubscribeURL** link to enable Amazon SNS notifications.

Note
In addition to the primary Lambda functions, this solution includes the `HelperFunction` Lambda function, which runs only during initial configuration or when resources are updated or deleted.
When running this solution, the `HelperFunction` function is inactive. However, do not delete the `HelperFunction` function as it is necessary to manage associated resources.

Step 2. Launch the Spoke Stack (Optional)

Use this procedure to launch the components necessary to monitor limits in secondary accounts. You must enter the secondary account IDs in the **Account List** parameter of the primary template before you launch this template in secondary accounts.

Note
You are responsible for the cost of the AWS services used while running this solution. See the Cost (p. 2) section for more details. For full details, see the pricing webpage for each AWS service you will be using in this solution.

1. Sign in to the AWS Management Console and click the button below to launch the `limit-monitor-spoke` 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.

   Note
   You must launch this solution in the US East (N. Virginia) Region.

3. On the **Create stack** page, verify that the correct template URL shows in the **Amazon S3 URL** text box and choose **Next**.
4. On the **Specify stack details** page, assign a name to your solution stack.
5. Under **Parameters**, review the parameter for the template, and modify it as necessary.

Parameter	Default	Description
Primary Account	*<Requires Input>*	The account ID of the primary account. The value must match the regular expression: `^\d{12}$`

6. Choose **Next**.
7. On the **Configure stack 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 stack** to deploy the stack.

   You can view the status of the stack in the AWS CloudFormation console in the Status column. You should see a status of **CREATE_COMPLETE** in approximately five minutes.

Step 3. Configure Slack Notifications (Optional)

Use this procedure to enable Slack notifications.

Note
Use this procedure if you specified parameters that did not already exist in AWS Systems Manager Parameter Store. If you specified parameters that already existed, you do not have to complete this step. For more information, see Slack Integration (p. 5).

1. Navigate to Slack's Incoming WebHooks app.
2. If necessary, log into Slack.
3. Select **Add Configuration**.
4. In the **Post to Channel** dropdown menu, choose a channel. Then, select **Add Incoming WebHooks integration**.
5. Copy the **WebHook URL**.
6. In the AWS Systems Manager console, under **Shared Resources** in the left pane, select **Parameter Store**.
7. Select the **Slack Hook URL Key** you provided during stack deployment, then select **Edit**
8. Replace the `SLACK_DUMMY` value with your WebHook URL and select **Save changes**.
9. Select the **Slack Channel Key** you provided during stack deployment, then select **Edit**
10. Replace the `SLACK_DUMMY` value with the channel you specified and select **Save changes**. For example, if your Slack channel name is `#limitmonitor`, enter `limitmonitor` as the value.

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 the AWS Security Center.

Additional Resources

AWS services documentation

- AWS Lambda
- AWS Trusted Advisor
- Amazon CloudWatch
- Amazon Simple Queue Service
- Amazon Simple Notification Service
- Amazon DynamoDB
- AWS Systems Manager
- AWS CloudFormation

Appendix A: Troubleshooting

The AWS Limit Monitor 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.

Level	Description
ERROR	Logs will include information on anything that causes an operation to fail.
WARNING	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.
INFO	Logs will include high-level information about how the function is operating. Logs will also include ERROR and WARN messages.
DEBUG	Logs will include information that might be helpful when debugging a problem with the function. Logs will also include ERROR, WARNING, and INFO messages.

Common Errors

Amazon CloudWatch Events Bus Permissions Error

If during spoke stack deployment, you received a **CREATE_FAILED** message for the `TAWarnRule` and/or the `TASErrorRule`, verify that the Amazon CloudWatch Events Bus in the primary account allows the spoke account to send events to the primary account.

Resolution

Update the primary stack with the secondary account ID or complete the following task:

1. In the primary account, navigate to the Amazon CloudWatch console.
2. In the navigation pane, select **Event Buses**.
3. Select **Add Permissions**.
4. For **Principal**, enter the applicable secondary account ID.
5. Select the **Everybody(*)** checkbox.
6. Choose **Add**.

Amazon DynamoDB Is Not Showing Events

If the solution's Amazon DynamoDB table is not updating or there are no messages in the solution's Amazon Simple Queue Service (Amazon SQS) queue, verify that the account IDs you entered in the solution's **Account List** parameter are comma-separated and use double quotation marks.

Resolution

Complete the following task:

1. In the primary account, navigate to the Amazon CloudWatch console.
2. In the navigation pane, select **Rules**.
3. Select the **TASNSRule** and verify the **account** parameter shows a comma-separated value inside double quotation marks.
4. Select the **TASQSRule** and verify the **account** parameter shows a comma-separated value inside double quotation marks.
5. Select the **TASlackRule** and verify the **account** parameter shows a comma-separated value inside double quotation marks.
6. If the **account** parameter in the **TASNSRule**, **TASQSRule**, or **TASlackRule** rules are not formatted correctly, update the stack with the correct format.
7. Choose **Add**.

Slack Notifications Are Not Being Received

If you do not receive Slack notifications for WARN or ERROR events, check the Amazon CloudWatch logs for an error message.

1. In the primary account, navigate to the Amazon CloudWatch console.
2. In the navigation pane, select **Logs**.
3. Select the /aws/lambda/*<stackname>*-SlackNotifier-*<randomstring>* Log Group.
4. Select the top (most recent) Log Stream.
5. Look for the following error.

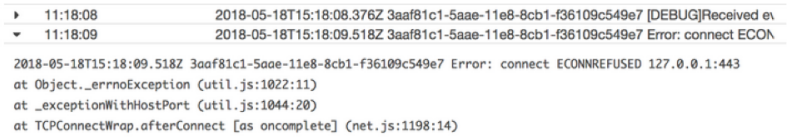

Figure 3: Slack notification error

Resolution

Complete the following task:

1. In the primary account, navigate to the AWS Systems Manager console.
2. In the navigation pane under **Shared Resources**, select **Parameter Store**.
3. Select the **SlackChannel** parameter and verify that the parameter shows the correct value.
4. Select the **SlackHookURL** parameter and verify that the parameter shows the correct value.

Email Notifications Are Not Being Received

If you do not receive email notifications, check to make sure that you have subscribed to the Amazon SNS topic.

1. In the primary account, navigate to the Amazon SNS console.

2. In the navigation pane, select **Topics**.
3. Select the *<stackname>*-SNSTopic-*<randomstring>* Amazon Resource Name (ARN) value.
4. Verify that the **Subscription ID** shows an ARN value.

Resolution

If the **Subscription ID** field shows `PendingConfirmation`, complete the following task:

1. Select the **checkbox** next to `PendingConfirmation`.
2. Under **Subscriptions**, select **Request Confirmations**.
3. Navigate to the applicable email inbox.
4. In the subscription notification email, select the **SubscribeURL** link.
5. In the Amazon SNS console, refresh and verify that the **Subscription ID** has an ARN value.

Appendix B: Customization

Change the Lambda Interval

By default, an AWS Lambda function runs once every 24 hours to refresh the AWS Trusted Advisor Service Limits checks. But, you can change how often the Lambda function is invoked by changing the mapping in the AWS CloudFormation template. Note that if you set the Lambda function to run more than once every 12 hours, you may experience a delay in your events arriving in Amazon DynamoDB.

To change the interval, modify the following mapping in the AWS CloudFormation template:

```
RefreshRate:
    CronSchedule:
      Default: rate(1 day)
```

For more information about acceptable rate expressions, see Rate Expressions in the Amazon CloudWatch Events User Guide.

Change the Monitored Services

By default, this solution checks all service limits that AWS Trusted Advisor checks. To specify which service limits you want to check, modify the following AWS CloudFormation template mapping.

```
EventsMap:
   Checks:
     Services:

 '"AutoScaling","CloudFormation","DynamoDB","EBS","EC2","ELB","IAM","Kinesis","RDS","Route53","SES","VP
```

Note that only the services you specify in the primary account will be checked in the secondary account(s). For example, if you specify only Amazon Elastic Compute Cloud (Amazon EC2) in your primary account, and Amazon EC2 and Amazon Elastic Block Store (Amazon EBS) in your secondary account, you will not receive status events for Amazon EBS from your secondary account because you have not specified Amazon EBS in your primary account.

If you specify Amazon EC2 and Amazon EBS in the primary account, and only Amazon EC2 in the secondary account, you will not receive status events for Amazon EBS from the secondary account because you have not specified Amazon EBS in your secondary account.

Appendix C: Collection of Operational Metrics

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

- **Solution ID:** The AWS solution identifier
- **Unique ID (UUID):** Randomly generated, unique identifier for each AWS Limit Monitor deployment
- **Timestamp:** Data-collection timestamp
- **SNS Events:** Whether Amazon Simple Notification Service notifications are enabled
- **Slack Events:** Whether Slack notifications are enabled
- **Spoke Count:** Count of the number of spoke accounts
- **Service:** The service that triggered the notification
- **Limit Name:** The resource that triggered the notification
- **Status Level:** The status level of the notification
- **Region:** The AWS Region where the resource is located

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, complete the following task:

Modify the AWS CloudFormation template mapping section as follows:

```
Mappings:
    Send-Data:
      SendAnonymousData: "Yes"
```

to

```
Mappings:
    Send-Data:
      SendAnonymousData: "No"
```

Source Code

Visit our GitHub repository to download the templates and scripts for this solution, and to share your customizations with others. The AWS Limit Monitor templates are generated using the AWS Cloud Development Kit (AWS CDK) (AWS CDK). Refer to the README.md file for more information.

Contributors

The following individuals contributed to this document:

- Garvit Singh
- Chaitanya Deolankar

Document Revisions

Date	Change
September 2016	Initial publication
February 2017	Solution updated to create two child AWS Lambda functions
June 2018	Added service-level granularity for AWS Trusted Advisor Service Limit checks; Amazon DynamoDB to store service utilization details; and Amazon CloudWatch Events Rule integration
July 2018	Added information about subscribing to the solution's Amazon SNS topic, and a sample Slack channel value for the solution's Parameter Store
April 2019	Added information about new parameters for the Slack incoming webhook URL and channel, and regular expressions for account IDs
July 2019	Added Amazon DynamoDB and Amazon Route 53 to the list of services AWS Trusted Advisor Service Limits checks
September 2019	Added information about Service Quota and vCPU limit checking functionality
December 2019	Added information on support for Node.js update
February 2020	Fixed a bug for the Amazon Simple Queue Service (Amazon SQS) server side encryption key and added AWS Trusted Advisor service limit checks
October 2020	Bug fixes; updated the AWS CloudFormation templates; for more information, refer to the CHANGELOG.md file in the GitHub repository

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.

AWS Limit Monitor is licensed under the terms of the of the Apache License Version 2.0 available at The Apache Software Foundation.