AppSentinels.ai
Application Security Re-invented

AppSentinels API Security Platform

Lambda Extension Deployment

Revision	Date Modified	Author	Comments
1.0	07-Aug-24	Arun	Initial spec for lambda deployment

Contents

AppSentinels Extension for AWS Lambda Functions

This document covers the steps to build and deploy the AppSentinels extension with AWS Lambda function. AppSentinels extensions captures the HTTP request headers and payload and forwards that to AppSentinels OnPrem Controller. Please note extensions cannot capture response payloads

Please note

- layer versions are made public, not the layer. This means that if a fix is done in AppSentinels extension, a new version is created with the fix and made public.
- A layer version is always accessed using layer ARN, even when it is made public.
- Contact support@appsentinels.ai for the latest ARN and the regions in which it is available.
- Current ARN:
    - arn:aws:lambda:ap-south-1:488922454646:layer:appsentinels-extension:44

Deploying Extension

Deploying Extension Manually

- Go to *Lambda > Functions > YourLambdaFunction*.
- Scroll down and click on **Add a layer**.
- Select **Specify an ARN**.
- Specify the ARN of the published AppSentinels extension.
- Click on **Add**.

Update Extension

If a newer version of AppSentinels extension is available, following steps can be used to upgrade to the newer version of the extension.

- Go to *Lambda > Functions > YourLambdaFunction*.
- Click on **Edit Layers**.
- Check the **Layer version** displayed along with the AppSentinels extension.
- Click on the version number and select the new version to deploy from the drop-down.
- Click on **Save**.

Automated Deployment of Extension

- Customer can deploy AppSentinels extension on single or all Lambda function using the script *deploy_extension.sh* https://sample-config.appsentinels.ai/appsentinels-deployment/aws-lambda/deploy_extension.sh
- This script requires following command line arguments:
    - AppSentinels layer ARN (AppSentinels shares this with customers).
    - Region (AppSentinels publishes the regions where it is available).
    - Lambda Function: all or name of the lambda function.

        - all: deploy extension on all the Lambda function in the given account in the given region.
        - lambda function name: deploy lambda function on specific lambda function with given name.
    - Controller URL (AppSentinels edge controller URL).
- Example: Deploy lambda extension on testHttpServer function.
  ./deploy_extension.sh "arn:aws:lambda:ap-south-1:488922454646:layer:appsentinels-extension:44" ap-south-1 testHttpServer controller.appsentinels.awstest.ai

Manage extensions

- The script *manage_extensions.sh* https://sample-config.appsentinels.ai/appsentinels-deployment/aws-lambda/manage_extensions.sh can be used to perform following operations:
    - List all the extension across all Lambda functions.
    - List all functions where an extension with given ARN is deployed.
    - Remove the extension with given ARN from all the functions.
- This script can be used to disable all instances of AppSentinels extension by running a single command.

Roles and permission requirements

- Lambda permissions
    - lambda:PublishLayerVersion
    - lambda:AddLayerVersionPermission
    - lambda:GetLayerVersion
    - lambda:UpdateFunctionConfiguration
    - lambda:GetFunction
    - lambda:DeleteLayerVersion
- Lambda function CloudWatch logs
    - logs:CreateLogGroup
    - logs:CreateLogStream
    - logs:PutLogEvents
    - logs:DescribeLogStreams
    - logs:GetLogEvents

Extension configuration

Following table lists the environment variables used to configure the AppSentinels extension (mandatory marked in green)

Environment Variable	Description
AS_EXTENSION_LOG_LEVEL	Log level - Debug/Info/Warn/Error/Fatal, default Info
AS_LOGGING_MODE	Logging mode - default/metadata/nologging
AS_CONTROLLER_ENDPOINT	AppSentinels Controller URL or IP address

Environment Variable	Description
AS_CONTROLLER_PORT	AppSentinels Controller port number
AS_MAX_BATCH_SIZE	Maximum number of logs in a batch (default 16)
AS_MAX_BODY_SIZE	Maximum request/response body size sent in the logs (128K)
AS_RELAY_PROTOCOL	Protocol use to relay the logs (http/https)
AS_SKIP_METHODS	HTTP methods skipped (default HEAD and OPTIONS)
AS_SKIP_URIS	Comma separated list of URIs to be skipped
AS_CONTENT_TYPE_REGEX	Regex defining content types process (json/xml/graphql/form)
AS_SERVER_CERT_VERIFY	Enable server certificate verification in HTTPS (default: no)

- Please note that AS_CONTROLLER_ENDPOINT is mandatory configuration parameter which needs to be specified during extension deployment.
- AS_LOGGING_MODE is set automatically by the extension based on the configuration.