# Trigger rescan of cloud connectors for new threats in Threat Center

First published - May 2024
Revision 3 – July 2024

**Overview:**
Customer requested script that will trigger a rescan of selected cloud connectors only if new threats were published to the Threat Center. This is expected to reduce the cost of unneeded scans, specifically for NonOS disk scanning, by allowing the NonOS disk scanning interval to be set to a longer period, while still ensuring that the environment is scanned for new threats soon after they are announced.

**Functionality:**
Create a script that performs the following on a scheduled basis:

1. Retrieve parameters from storage – connection credentials, last scan date, list of connectors to scan, etc.
2. Get the date of the latest non-informational threat published to the Threat Center
3. Get the date of the most recent triggered scan
4. If the latest threat is newer
    a. Update the most recent triggered scan date
    b. Request a scan of the connectors in the supplied list

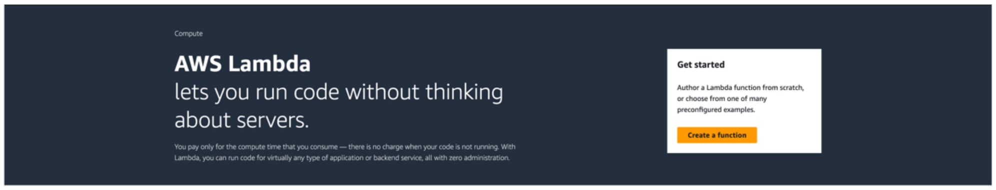
**Implementation:**
The script is implemented as an AWS Lambda function containing a single python file which is schedule to run on a desired repeating basis (e.g. daily). The Lambda function makes use of an SSM Parameter store to retrieve the following parameters. The function also updates the lastScanDate field based on the logic above:

| Parameter | Purpose | Value |
|---|---|---|
| Client_id | Wiz Service Account ID | Populate from Wiz UI when creating Service Account |
| Client_secret | Wiz Service Account Secret | Populate from Wiz UI when creating Service Account |
| Connector_list | List of connectors to rescan | Comma-separated list, spaces allowed for readability |
| debugMode | Detailed debugging mode | True or False (default False) |
| lastScanDate | Date of last triggered scan | Must contain an initial value, will be rewritten with the current date if invalid. |

## Installation and Setup:

### AWS Lambda Setup:

1. Download the WizThreatScan.zip and python_package.zip files (git clone https://github.com/jsing3r/WizThreatScan.git)
2. Login to AWS console with Admin rights (create roles, assign role, create SSM, create Lambda function, etc.)
3. Set your desired region (e.g. us-east-1)
4. Create a new Lambda function



Compute

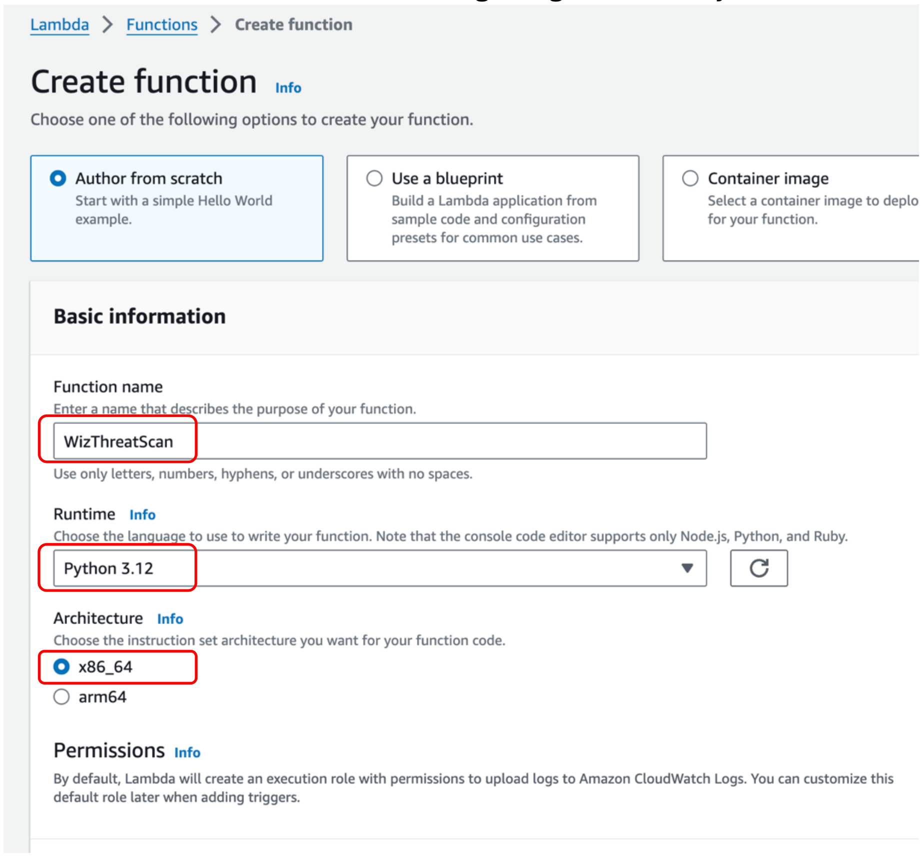
**AWS Lambda**
lets you run code without thinking about servers.

You pay only for the compute time that you consume — there is no charge when your code is not running. With Lambda, you can run code for virtually any type of application or backend service, all with zero administration.

**Get started**

Author a Lambda function from scratch, or choose from one of many preconfigured examples.

5. Use the information in the following image to create your Lambda function:



Create function Info

Choose one of the following options to create your function.

[✓ Author from scratch] Start with a simple Hello World example.

Use a blueprint — Build a Lambda application from sample code and configuration presets for common use cases.

Container image — Select a container image to deplo for your function.

**Basic information**

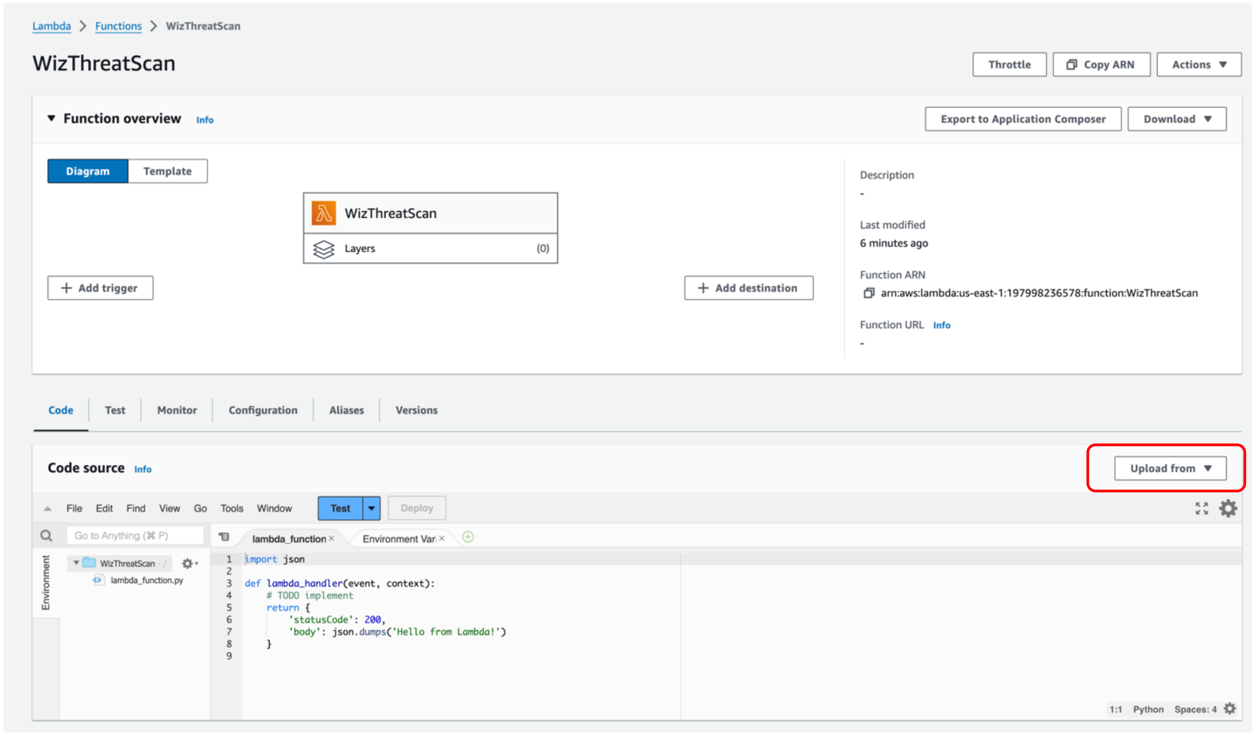
Function name
Enter a name that describes the purpose of your function.
WizThreatScan
Use only letters, numbers, hyphens, or underscores with no spaces.

Runtime Info
Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.
Python 3.12

Architecture Info
Choose the instruction set architecture you want for your function code.
[✓ x86_64]
arm64

Permissions Info
By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.

6. You should have a default Lambda function created as shown below:



Lambda > Functions > WizThreatScan

WizThreatScan

Function overview

Description
-

Last modified
6 minutes ago

Function ARN
arn:aws:lambda:us-east-1:197998236578:function:WizThreatScan

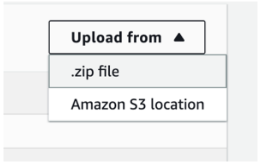
Function URL
-

Code source

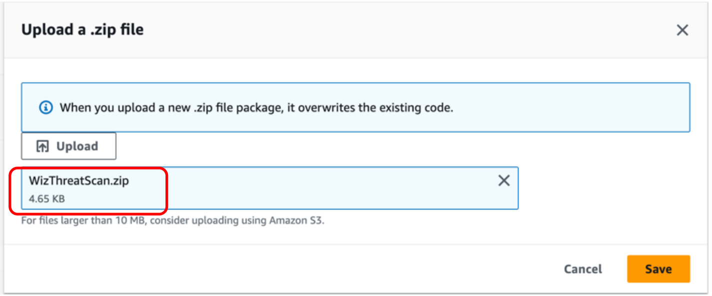
```
import json

def lambda_handler(event, context):
    # TODO implement
    return {
        'statusCode': 200,
        'body': json.dumps('Hello from Lambda!')
    }
```

7. Select Upload from option and select .zip file



8. Select the WizThreatScan.zip file saved earlier to update the Lambda code



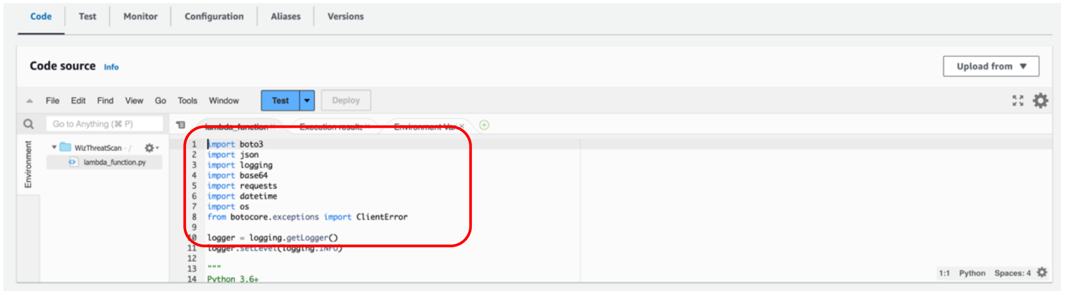
Upload a .zip file

When you upload a new .zip file package, it overwrites the existing code.

WizThreatScan.zip
4.65 KB

For files larger than 10 MB, consider uploading using Amazon S3.

9. Confirm that the lambda_function.py has been updated as shown below

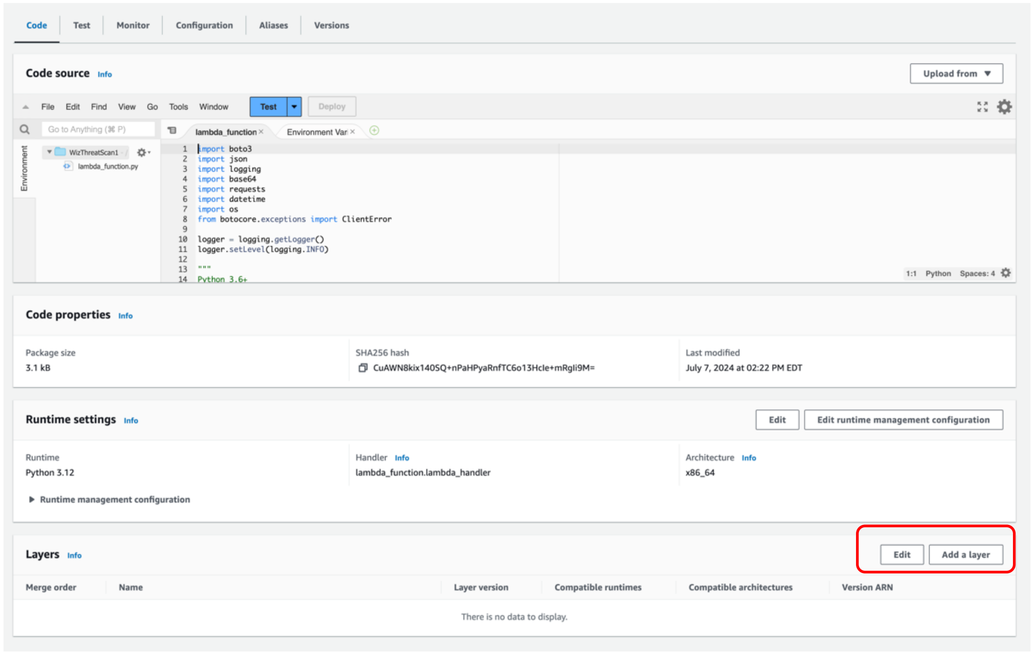


```
import boto3
import json
import logging
import base64
import requests
import datetime
import os
from botocore.exceptions import ClientError

logger = logging.getLogger()
logger.setLevel(logging.INFO)

"""
Python 3.6+
```

10. Scroll down to the Layers section on the Lambda page



Code source Info

WizThreatScan1
lambda_function.py

lambda_function × Environment Var ×

```
import boto3
import json
import logging
import base64
import requests
import datetime
import os
from botocore.exceptions import ClientError

logger = logging.getLogger()
logger.setLevel(logging.INFO)

"""
Python 3.6+
```

Code properties Info

| Package size | SHA256 hash | Last modified |
|---|---|---|
| 3.1 kB | CuAWN8kix140SQ+nPaHPyaRnfTC6o13Hcle+mRgli9M= | July 7, 2024 at 02:22 PM EDT |

Runtime settings Info

| Runtime | Handler Info | Architecture Info |
|---|---|---|
| Python 3.12 | lambda_function.lambda_handler | x86_64 |

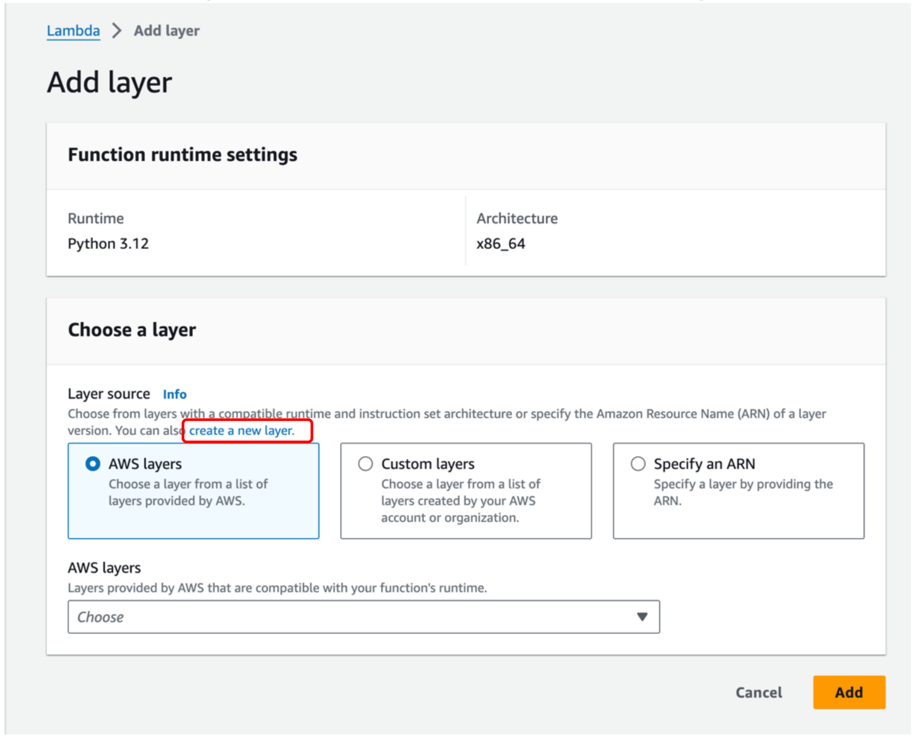
▶ Runtime management configuration

Layers Info

| Merge order | Name | Layer version | Compatible runtimes | Compatible architectures | Version ARN |
|---|---|---|---|---|---|

There is no data to display.

11. Select Add a layer, and then select **create a new layer**



Add layer

Function runtime settings

| Runtime | Architecture |
|---|---|
| Python 3.12 | x86_64 |

Choose a layer

Layer source Info

Choose from layers with a compatible runtime and instruction set architecture or specify the Amazon Resource Name (ARN) of a layer version. You can also create a new layer.

[✓ AWS layers] Choose a layer from a list of layers provided by AWS.

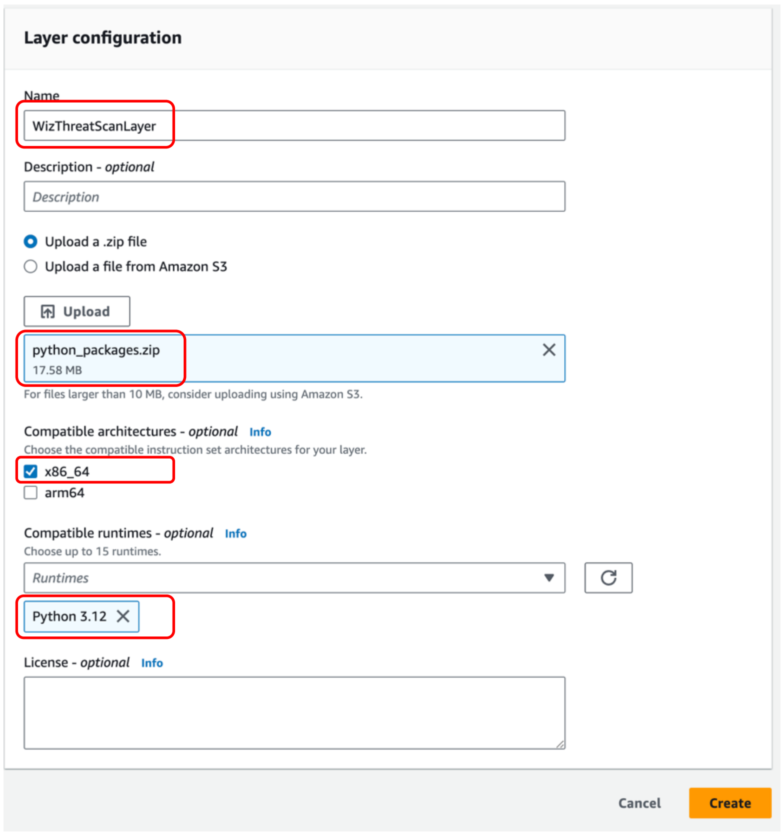
Custom layers — Choose a layer from a list of layers created by your AWS account or organization.

Specify an ARN — Specify a layer by providing the ARN.

AWS layers

Layers provided by AWS that are compatible with your function's runtime.

12. Create a new layer called WizThreatScanLayer and upload the python_packages.zip file to the layer. Also select the compatible architecture and runtimes as shown



**Layer configuration**

Name
WizThreatScanLayer

Description - *optional*
*Description*

[✓ Upload a .zip file]
Upload a file from Amazon S3

Upload
python_packages.zip
17.58 MB
For files larger than 10 MB, consider uploading using Amazon S3.

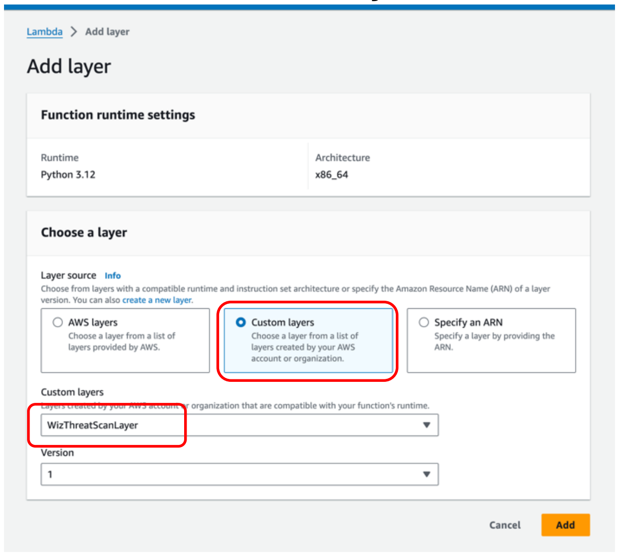
Compatible architectures - *optional* Info
Choose the compatible instruction set architectures for your layer.
[✓ x86_64]
arm64

Compatible runtimes - *optional* Info
Choose up to 15 runtimes.
*Runtimes*
Python 3.12

License - *optional* Info

Cancel Create

13. Return to the Add a layer dialog, select Custom layers, then select the newly created **WizThreatScanLayer** and Version 1 in the pulldown list:



Lambda > Add layer

Add layer

**Function runtime settings**

| | |
|---|---|
| Runtime | Architecture |
| Python 3.12 | x86_64 |

**Choose a layer**

Layer source Info
Choose from layers with a compatible runtime and instruction set architecture or specify the Amazon Resource Name (ARN) of a layer version. You can also create a new layer.

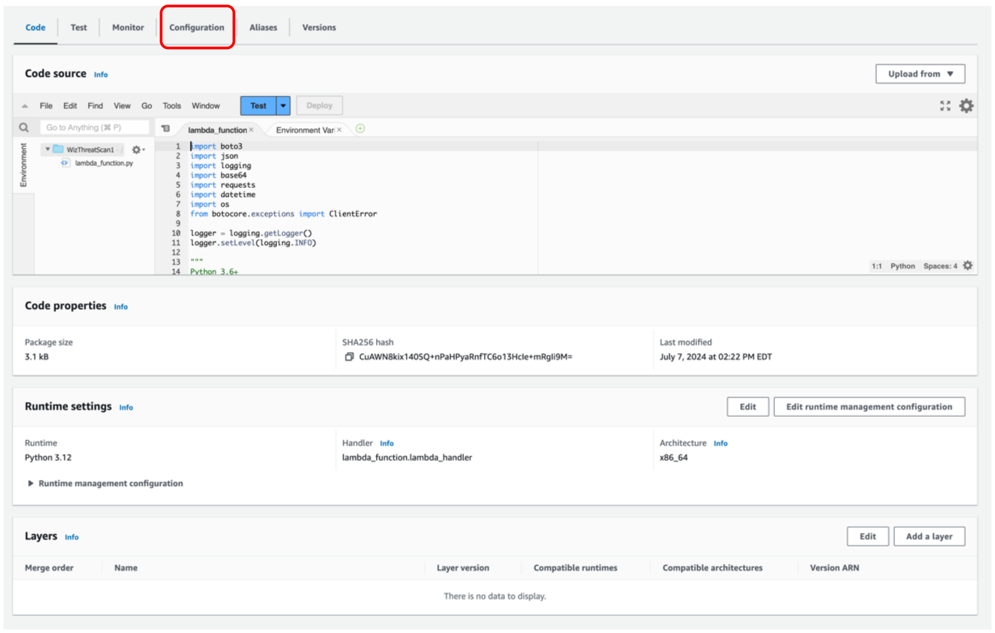
AWS layers – Choose a layer from a list of layers provided by AWS.
[✓ Custom layers – Choose a layer from a list of layers created by your AWS account or organization.]
Specify an ARN – Specify a layer by providing the ARN.

Custom layers
Layers created by your AWS account or organization that are compatible with your function's runtime.
WizThreatScanLayer

Version
1

Cancel Add

14. Navigate to the **Configuration** tab



```
import boto3
import json
import logging
import base64
import requests
import datetime
import os
from botocore.exceptions import ClientError

logger = logging.getLogger()
logger.setLevel(logging.INFO)

"""
Python 3.6+
```

**Code properties** Info

| Package size | SHA256 hash | Last modified |
|---|---|---|
| 3.1 kB | CuAWN8kix140SQ+nPaHPyaRnfTC6o13Hcle+mRgli9M= | July 7, 2024 at 02:22 PM EDT |

**Runtime settings** Info

| Runtime | Handler Info | Architecture Info |
|---|---|---|
| Python 3.12 | lambda_function.lambda_handler | x86_64 |

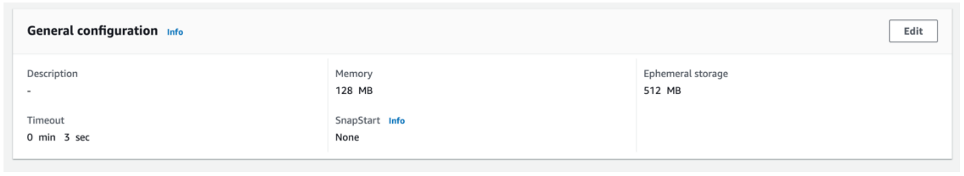
▶ Runtime management configuration

**Layers** Info

| Merge order | Name | Layer version | Compatible runtimes | Compatible architectures | Version ARN |
|---|---|---|---|---|---|

There is no data to display.

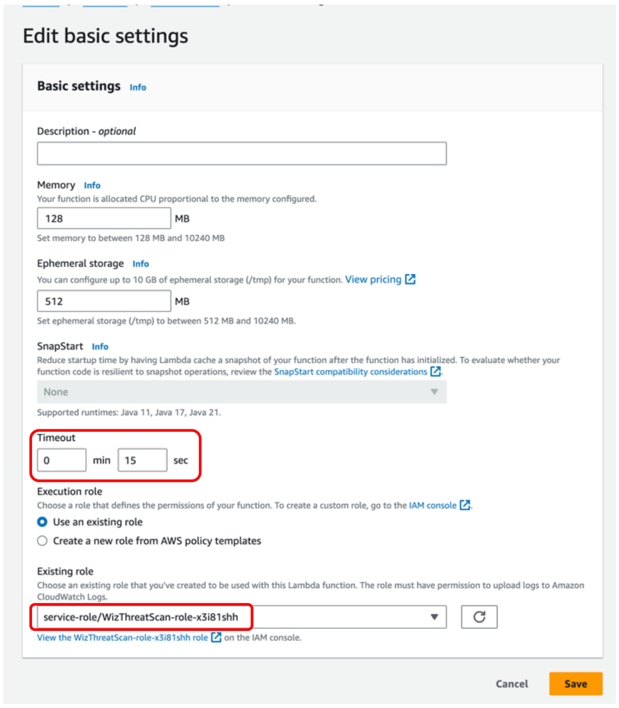
15. Under General configuration, select Edit



**General configuration** Info

| | | |
|---|---|---|
| Description<br>- | Memory<br>128 MB | Ephemeral storage<br>512 MB |
| Timeout<br>0 min 3 sec | SnapStart Info<br>None | |

16. Set the Timeout value to 15 sec instead of the default of 3 sec. Also make note of role that will be used to execute the Lambda function



Edit basic settings

**Basic settings** Info

Description - *optional*

Memory Info
Your function is allocated CPU proportional to the memory configured.
128 MB
Set memory to between 128 MB and 10240 MB

Ephemeral storage Info
You can configure up to 10 GB of ephemeral storage (/tmp) for your function. View pricing
512 MB
Set ephemeral storage (/tmp) to between 512 MB and 10240 MB.

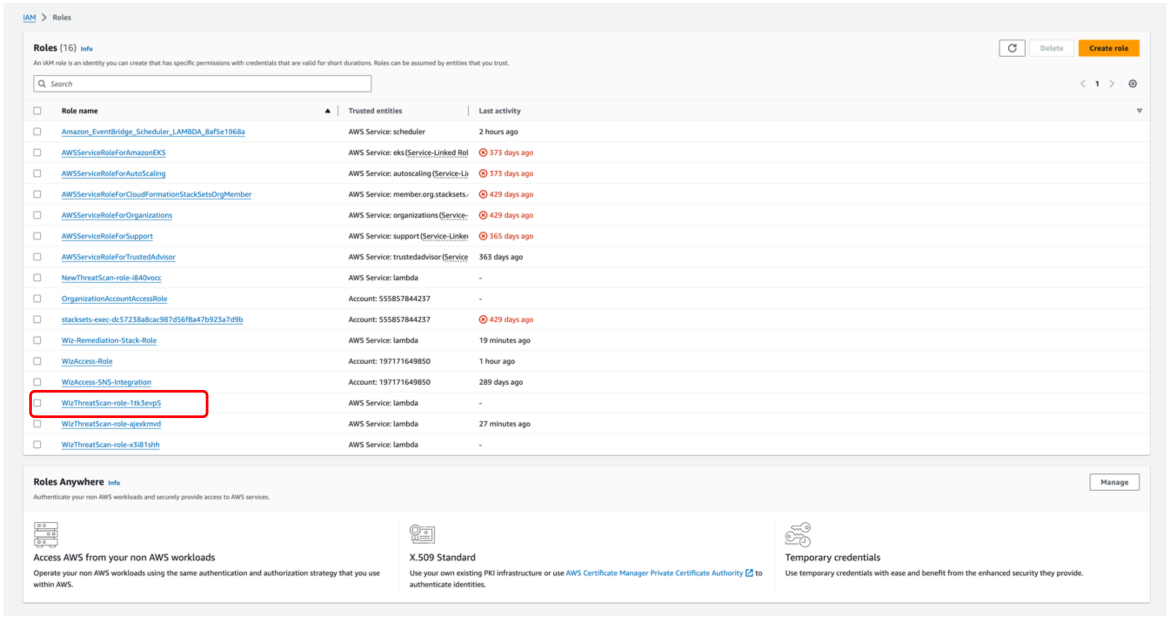
SnapStart Info
Reduce startup time by having Lambda cache a snapshot of your function after the function has initialized. To evaluate whether your function code is resilient to snapshot operations, review the SnapStart compatibility considerations.
None
Supported runtimes: Java 11, Java 17, Java 21.

Timeout
0 min 15 sec

Execution role
Choose a role that defines the permissions of your function. To create a custom role, go to the IAM console.
● Use an existing role
○ Create a new role from AWS policy templates

Existing role
Choose an existing role that you've created to be used with this Lambda function. The role must have permission to upload logs to Amazon CloudWatch Logs.
service-role/WizThreatScan-role-x3i81shh
View the WizThreatScan-role-x3i81shh role on the IAM console.

17. Navigate to IAM in the AWS console, and select Roles. Locate the **WizThreatScan** role identified above:



IAM > Roles

**Roles** (16) Info

An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.

| Role name | Trusted entities | Last activity |
|---|---|---|
| Amazon_EventBridge_Scheduler_LAMBDA_8af5e1968a | AWS Service: scheduler | 2 hours ago |
| AWSServiceRoleForAmazonEKS | AWS Service: eks (Service-Linked Rol | 373 days ago |
| AWSServiceRoleForAutoScaling | AWS Service: autoscaling (Service-Li | 373 days ago |
| AWSServiceRoleForCloudFormationStackSetsOrgMember | AWS Service: member.org.stacksets. | 429 days ago |
| AWSServiceRoleForOrganizations | AWS Service: organizations (Service- | 429 days ago |
| AWSServiceRoleForSupport | AWS Service: support (Service-Linke | 365 days ago |
| AWSServiceRoleForTrustedAdvisor | AWS Service: trustedadvisor (Service | 363 days ago |
| NewThreatScan-role-i840vocc | AWS Service: lambda | - |
| OrganizationAccountAccessRole | Account: 555857844237 | - |
| stacksets-exec-dc57238a8cac987d56f8a47b923a7d9b | Account: 555857844237 | 429 days ago |
| Wiz-Remediation-Stack-Role | AWS Service: lambda | 19 minutes ago |
| WizAccess-Role | Account: 197171649850 | 1 hour ago |
| WizAccess-SNS-Integration | Account: 197171649850 | 289 days ago |
| WizThreatScan-role-1tk3evp5 | AWS Service: lambda | - |
| WizThreatScan-role-ajexkmvd | AWS Service: lambda | 27 minutes ago |
| WizThreatScan-role-x3i81shh | AWS Service: lambda | - |

**Roles Anywhere** Info

Authenticate your non AWS workloads and securely provide access to AWS services.

Access AWS from your non AWS workloads

Operate your non AWS workloads using the same authentication and authorization strategy that you use within AWS.

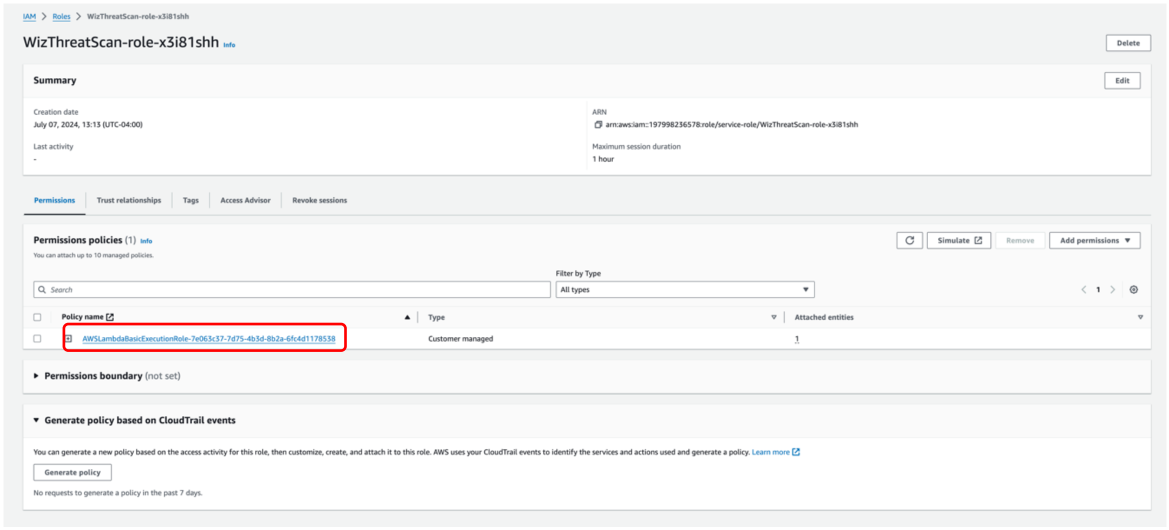
X.509 Standard

Use your own existing PKI infrastructure or use AWS Certificate Manager Private Certificate Authority to authenticate identities.

Temporary credentials

Use temporary credentials with ease and benefit from the enhanced security they provide.

18. Select the **WizThreatScan** role and confirm the permission policies are like that shown below



IAM > Roles > WizThreatScan-role-x3i81shh

WizThreatScan-role-x3i81shh Info

**Summary**

Creation date
July 07, 2024, 13:13 (UTC-04:00)

Last activity
-

ARN
arn:aws:iam::197998236578:role/service-role/WizThreatScan-role-x3i81shh

Maximum session duration
1 hour

**Permissions policies** (1) Info

You can attach up to 10 managed policies.

Filter by Type: All types

| Policy name | Type | Attached entities |
|---|---|---|
| AWSLambdaBasicExecutionRole-7e063c37-7d75-4b3d-8b2a-6fc4d1178538 | Customer managed | 1 |

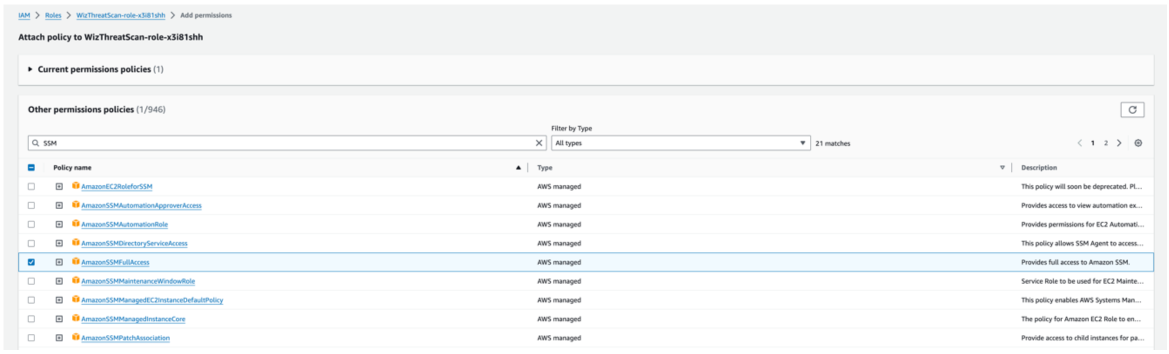
**Permissions boundary** (not set)

**Generate policy based on CloudTrail events**

You can generate a new policy based on the access activity for this role, then customize, create, and attach it to this role. AWS uses your CloudTrail events to identify the services and actions used and generate a policy. Learn more

No requests to generate a policy in the past 7 days.

19. Select Add permissions and Attach policy, then select **AmazonSSMFullAccess** policy



IAM > Roles > WizThreatScan-role-x3i81shh > Add permissions

Attach policy to WizThreatScan-role-x3i81shh

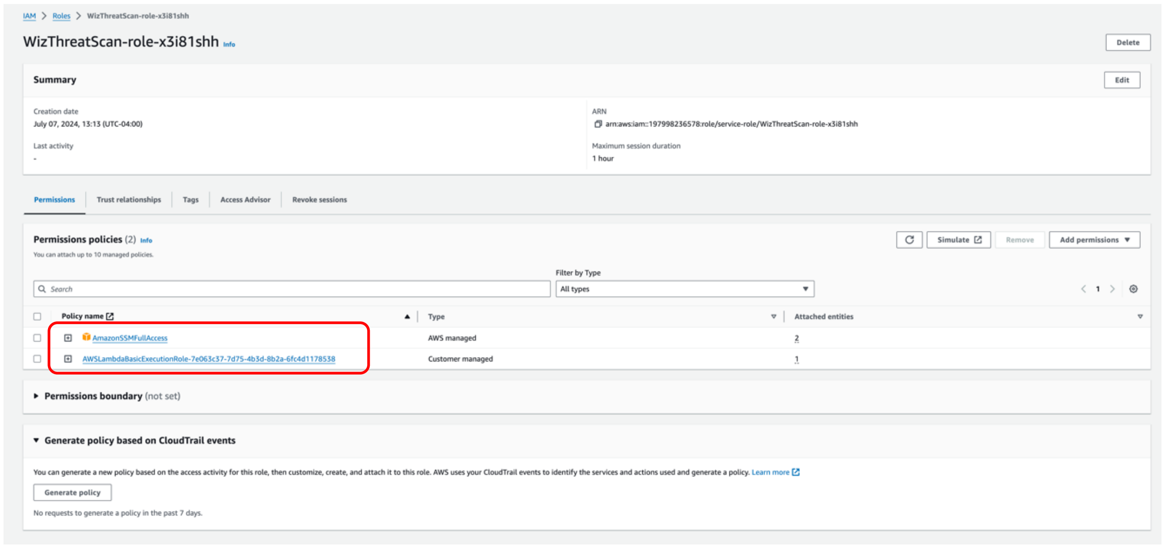
Current permissions policies (1)

Other permissions policies (1/946)

Search: SSM — Filter by Type: All types — 21 matches

| | Policy name | Type | Description |
|---|---|---|---|
| ☐ | AmazonEC2RoleforSSM | AWS managed | This policy will soon be deprecated. Pl... |
| ☐ | AmazonSSMAutomationApproverAccess | AWS managed | Provides access to view automation ex... |
| ☐ | AmazonSSMAutomationRole | AWS managed | Provides permissions for EC2 Automati... |
| ☐ | AmazonSSMDirectoryServiceAccess | AWS managed | This policy allows SSM Agent to access... |
| ✓ | AmazonSSMFullAccess | AWS managed | Provides full access to Amazon SSM. |
| ☐ | AmazonSSMMaintenanceWindowRole | AWS managed | Service Role to be used for EC2 Mainte... |
| ☐ | AmazonSSMManagedEC2InstanceDefaultPolicy | AWS managed | This policy enables AWS Systems Man... |
| ☐ | AmazonSSMManagedInstanceCore | AWS managed | The policy for Amazon EC2 Role to en... |
| ☐ | AmazonSSMPatchAssociation | AWS managed | Provide access to child instances for pa... |

20. Confirm that the WizThreatScan role now has the **AmazonSSMFullAccess** policy attached.



IAM > Roles > WizThreatScan-role-x3i81shh

WizThreatScan-role-x3i81shh

Summary

Creation date: July 07, 2024, 13:13 (UTC-04:00)

ARN: arn:aws:iam::197998236578:role/service-role/WizThreatScan-role-x3i81shh

Last activity: -

Maximum session duration: 1 hour

Permissions | Trust relationships | Tags | Access Advisor | Revoke sessions

Permissions policies (2)

You can attach up to 10 managed policies.

Filter by Type: All types

| Policy name | Type | Attached entities |
|---|---|---|
| AmazonSSMFullAccess | AWS managed | 2 |
| AWSLambdaBasicExecutionRole-7e063c37-7d75-4b3d-8b2a-6fc4d1178538 | Customer managed | 1 |

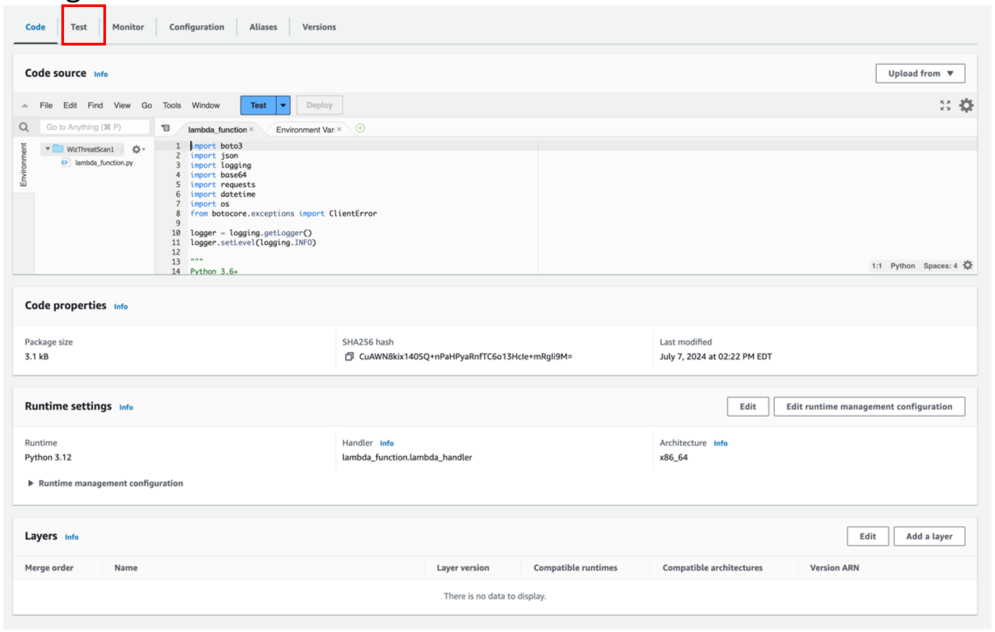
Permissions boundary (not set)

Generate policy based on CloudTrail events

You can generate a new policy based on the access activity for this role, then customize, create, and attach it to this role. AWS uses your CloudTrail events to identify the services and actions used and generate a policy. Learn more

No requests to generate a policy in the past 7 days.

21. Navigate to the Test tab



Code | Test | Monitor | Configuration | Aliases | Versions

Code source

```
import boto3
import json
import logging
import base64
import requests
import datetime
import os
from botocore.exceptions import ClientError

logger = logging.getLogger()
logger.setLevel(logging.INFO)

"""
Python 3.6+
```

Code properties

Package size: 3.1 kB

SHA256 hash: CuAWN8kix140SQ+nPaHPyaRnfTC6o13Hcle+mRgli9M=

Last modified: July 7, 2024 at 02:22 PM EDT

Runtime settings

Runtime: Python 3.12

Handler: lambda_function.lambda_handler

Architecture: x86_64

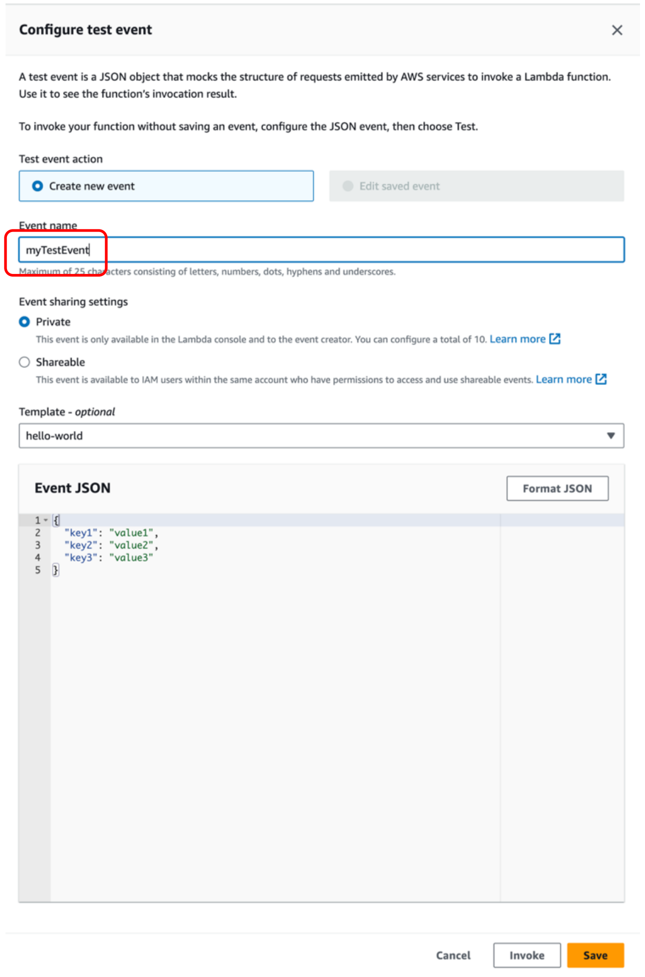
Runtime management configuration

Layers

| Merge order | Name | Layer version | Compatible runtimes | Compatible architectures | Version ARN |
|---|---|---|---|---|---|

There is no data to display.

22. Create a new test event called **myTestEvent** as shown. This test data will not be used but it allows the Test button to be used to test our function.



Configure test event

A test event is a JSON object that mocks the structure of requests emitted by AWS services to invoke a Lambda function. Use it to see the function's invocation result.

To invoke your function without saving an event, configure the JSON event, then choose Test.

Test event action: [✓ Create new event] Edit saved event

Event name: myTestEvent

Maximum of 25 characters consisting of letters, numbers, dots, hyphens and underscores.

Event sharing settings: [✓ Private] Shareable

Private: This event is only available in the Lambda console and to the event creator. You can configure a total of 10. Learn more

Shareable: This event is available to IAM users within the same account who have permissions to access and use shareable events. Learn more

Template - *optional*: hello-world

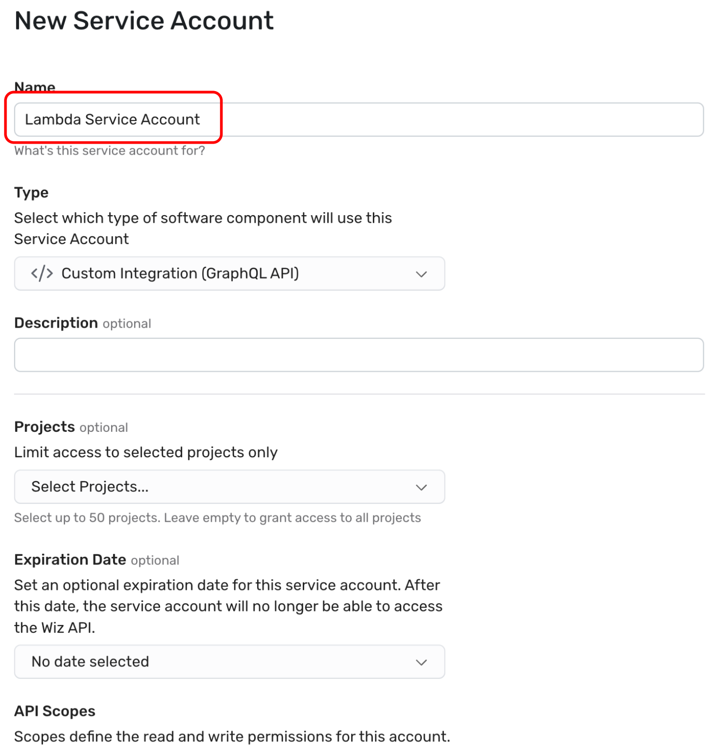
Event JSON

```
{
  "key1": "value1",
  "key2": "value2",
  "key3": "value3"
}
```

## Wiz UI Setup:

1. Navigate to Wiz – Settings / Access Management / Service Accounts

2. Create a new Service Account of type Custom Integration (GraphQL API)



New Service Account

Name
Lambda Service Account
What's this service account for?

Type
Select which type of software component will use this Service Account
Custom Integration (GraphQL API)

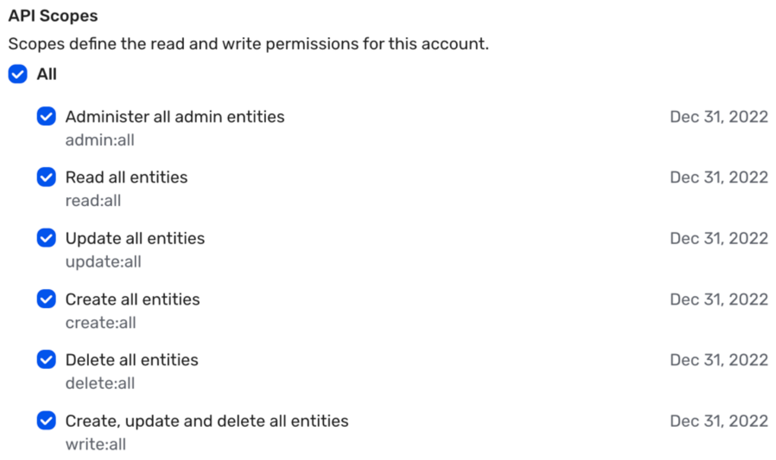
Description optional

Projects optional
Limit access to selected projects only
Select Projects...
Select up to 50 projects. Leave empty to grant access to all projects

Expiration Date optional
Set an optional expiration date for this service account. After this date, the service account will no longer be able to access the Wiz API.
No date selected

API Scopes
Scopes define the read and write permissions for this account.

3. Select all API Scopes

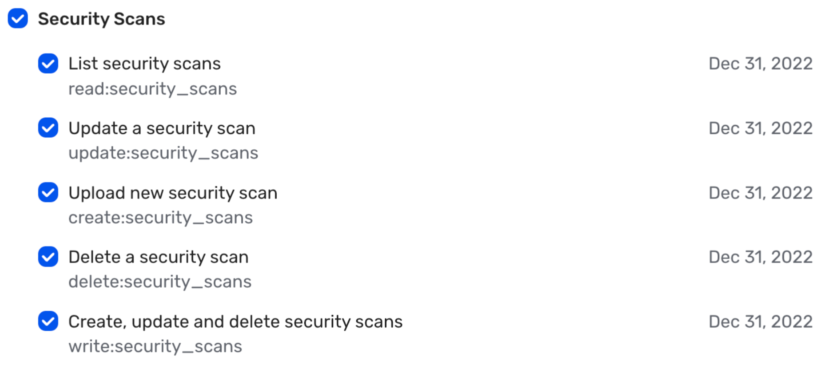


API Scopes
Scopes define the read and write permissions for this account.

✓ All

| | |
|---|---|
| ✓ Administer all admin entities admin:all | Dec 31, 2022 |
| ✓ Read all entities read:all | Dec 31, 2022 |
| ✓ Update all entities update:all | Dec 31, 2022 |
| ✓ Create all entities create:all | Dec 31, 2022 |
| ✓ Delete all entities delete:all | Dec 31, 2022 |
| ✓ Create, update and delete all entities write:all | Dec 31, 2022 |

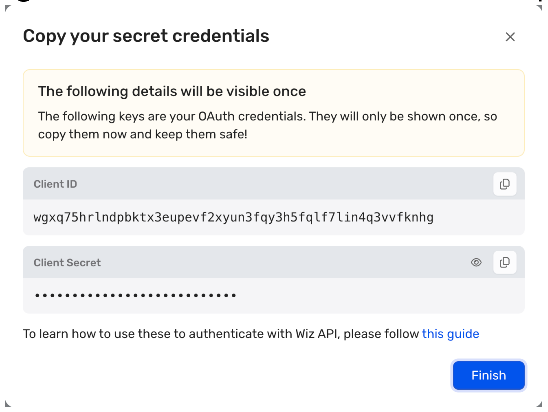
4. Select all Security Scan Scopes



✓ Security Scans

| | |
|---|---|
| ✓ List security scans read:security_scans | Dec 31, 2022 |
| ✓ Update a security scan update:security_scans | Dec 31, 2022 |
| ✓ Upload new security scan create:security_scans | Dec 31, 2022 |
| ✓ Delete a security scan delete:security_scans | Dec 31, 2022 |
| ✓ Create, update and delete security scans write:security_scans | Dec 31, 2022 |

5. Copy your secret credentials to a secure location as they will not be displayed again. These will be entered in the SSM parameter list


Copy your secret credentials

The following details will be visible once

The following keys are your OAuth credentials. They will only be shown once, so copy them now and keep them safe!

Client ID

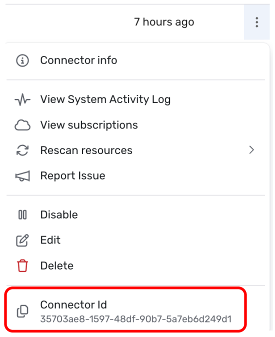
wgxq75hrlndpbktx3eupevf2xyun3fqy3h5fqlf7lin4q3vvfknhg

Client Secret

•••••••••••••••••••••••••••••

To learn how to use these to authenticate with Wiz API, please follow this guide

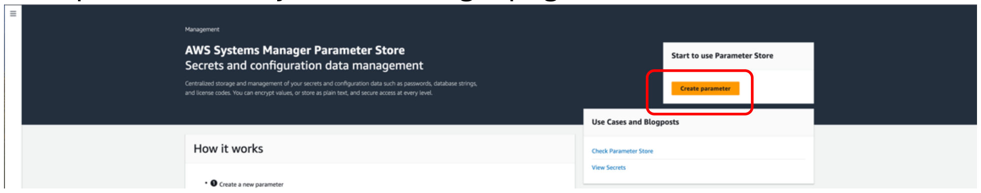
6. Open the Wiz Settings / Deployments page and identify the connectors that should be rescanned for new threats. Click the menu at the end of the connector line to find the connector ID. These IDs will be entered as comma-separated values in the SSM parameter list.



Connector Id
35703ae8-1597-48df-90b7-5a7eb6d249d1

**AWS SSM Setup:**

1. Open the AWS Systems Manager page and select Parameter Store


Management

AWS Systems Manager Parameter Store
Secrets and configuration data management

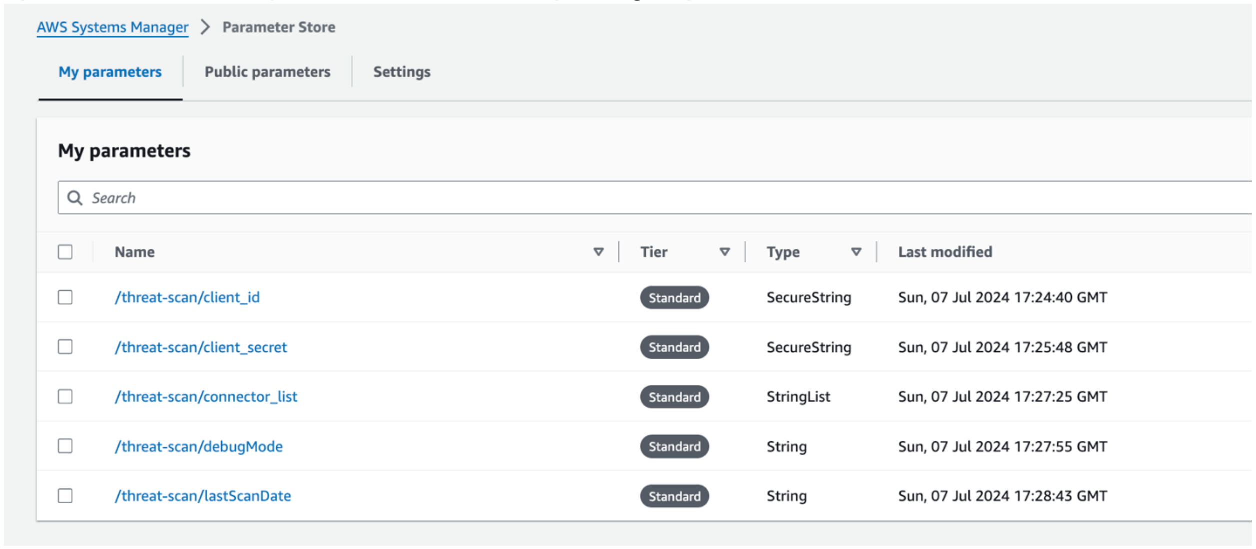
Centralized storage and management of your secrets and configuration data such as passwords, database strings, and license codes. You can encrypt values, or store as plain text, and secure access at every level.

2. Create the following parameters :

| **Parameter** | **Type** | **Value** |
|---|---|---|
| /threat-scan/client_id | SecureString | Client_id from Wiz Service Account created above |
| /threat-scan/client_secret | SecureString | Client_secret from Wiz Service Account created above |
| /threat-scan/connector_list | StringList | Comma-separated list of connector IDs copied above, spaces allowed for readability |

| | | |
|---|---|---|
| /threat-scan/debugMode | String | True or False (Set to True for detailed debugging information) |
| /threat-scan/lastScanDate | String | Must contain some initial value which will be rewritten with the current date if invalid. A good starting example is NEVER. |

3. Ensure that the following parameters are created and have the correct values as specified above. Pay attention to exact spelling of parameters:


AWS Systems Manager > Parameter Store

My parameters | Public parameters | Settings

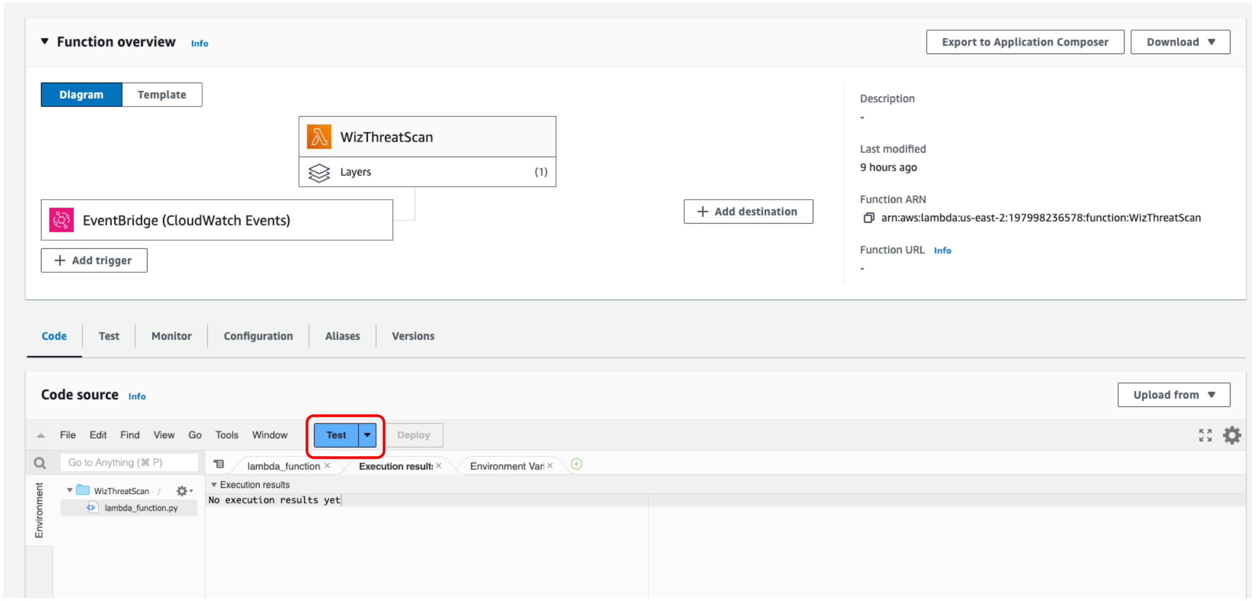
**My parameters**

Search

| Name | Tier | Type | Last modified |
|---|---|---|---|
| /threat-scan/client_id | Standard | SecureString | Sun, 07 Jul 2024 17:24:40 GMT |
| /threat-scan/client_secret | Standard | SecureString | Sun, 07 Jul 2024 17:25:48 GMT |
| /threat-scan/connector_list | Standard | StringList | Sun, 07 Jul 2024 17:27:25 GMT |
| /threat-scan/debugMode | Standard | String | Sun, 07 Jul 2024 17:27:55 GMT |
| /threat-scan/lastScanDate | Standard | String | Sun, 07 Jul 2024 17:28:43 GMT |

**Testing the script:**

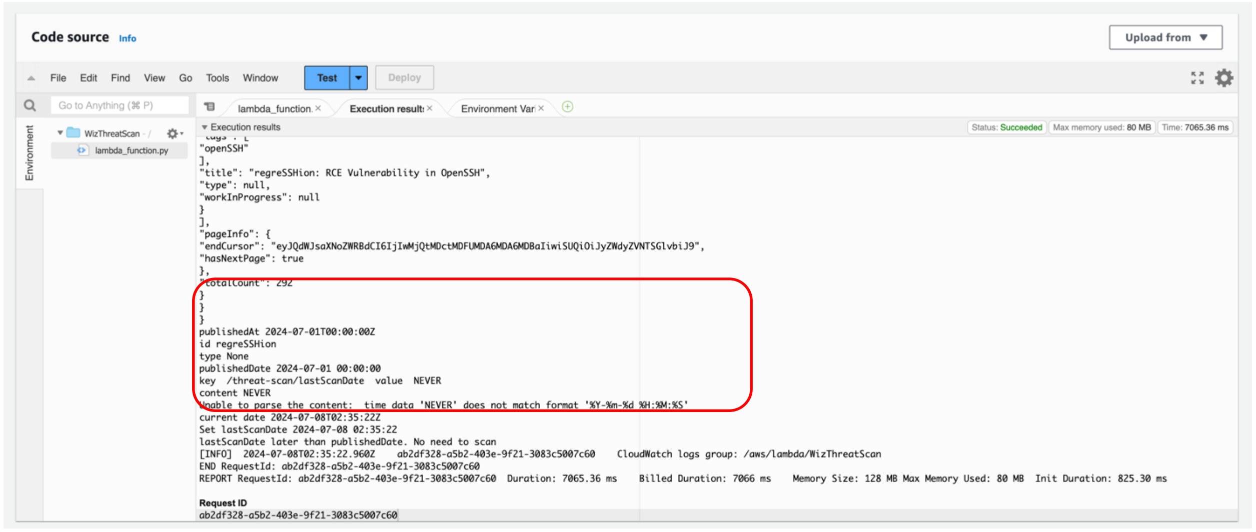
1. Configure all setup elements in AWS and Wiz as described above
2. Select Test button



Description
-
Last modified
9 hours ago
Function ARN
arn:aws:lambda:us-east-2:197998236578:function:WizThreatScan
Function URL Info
-

Execution results
```
No execution results yet
```

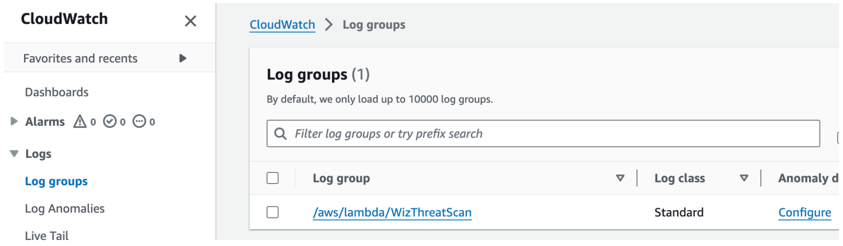
3. Observe output in execution log window. For first invocation, lastScanDate (set to NEVER in this example) is overwritten with the current date. For every following run, the lastScanDate will be compared with the date of the most recent threat center notice, and a rescan will be triggered as required.



```
"openSSH"
],
"title": "regreSSHion: RCE Vulnerability in OpenSSH",
"type": null,
"workInProgress": null
}
],
"pageInfo": {
"endCursor": "eyJQdWJsaXNoZWRBdCI6IjIwMjQtMDctMDFUMDA6MDA6MDBaIiwiSUQiOiJyZWdyZVNTSGlvbiJ9",
"hasNextPage": true
},
"totalCount": 292
}
}
}
publishedAt 2024-07-01T00:00:00Z
id regreSSHion
type None
publishedDate 2024-07-01 00:00:00
key  /threat-scan/lastScanDate  value  NEVER
content NEVER
Unable to parse the content:  time data 'NEVER' does not match format '%Y-%m-%d %H:%M:%S'
current date 2024-07-08T02:35:22Z
Set lastScanDate 2024-07-08 02:35:22
lastScanDate later than publishedDate. No need to scan
[INFO]  2024-07-08T02:35:22.960Z    ab2df328-a5b2-403e-9f21-3083c5007c60    CloudWatch logs group: /aws/lambda/WizThreatScan
END RequestId: ab2df328-a5b2-403e-9f21-3083c5007c60
REPORT RequestId: ab2df328-a5b2-403e-9f21-3083c5007c60  Duration: 7065.36 ms    Billed Duration: 7066 ms    Memory Size: 128 MB Max Memory Used: 80 MB  Init Duration: 825.30 ms

Request ID
ab2df328-a5b2-403e-9f21-3083c5007c60
```

4. Verify CloudWatch log group for the Lambda function

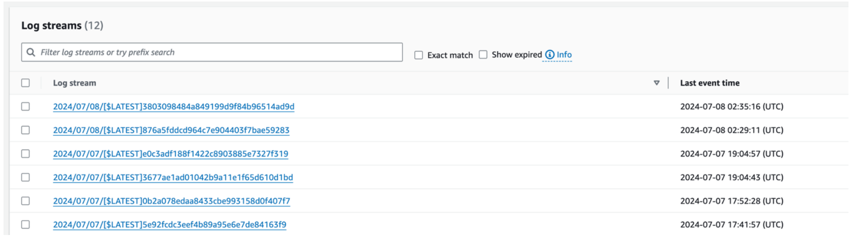


**Log groups** (1)

By default, we only load up to 10000 log groups.

| Log group | Log class | Anomaly d |
|---|---|---|
| /aws/lambda/WizThreatScan | Standard | Configure |

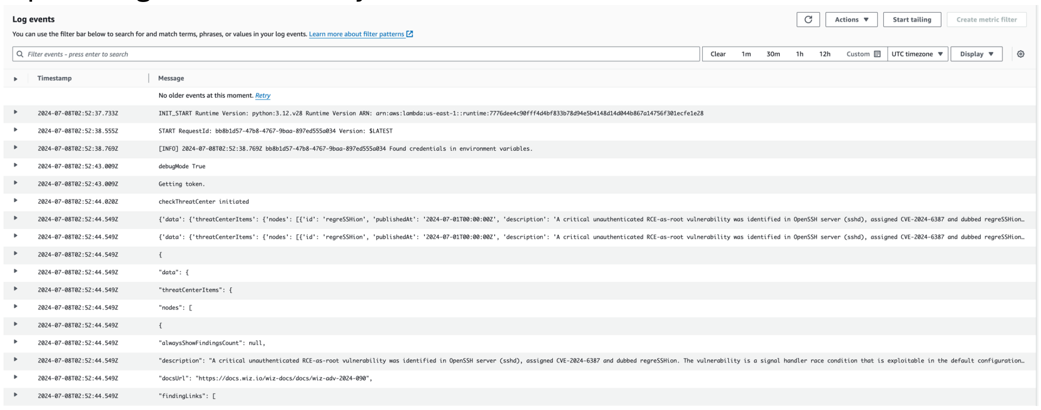
5. Examine CloudWatch logs for the execution of the Lambda function



**Log streams** (12)

| Log stream | Last event time |
|---|---|
| 2024/07/08/[$LATEST]3803098484a849199d9f84b96514ad9d | 2024-07-08 02:35:16 (UTC) |
| 2024/07/08/[$LATEST]876a5fddcd964c7e904403f7bae59283 | 2024-07-08 02:29:11 (UTC) |
| 2024/07/07/[$LATEST]e0c3adf188f1422c8903885e7327f319 | 2024-07-07 19:04:57 (UTC) |
| 2024/07/07/[$LATEST]3677ae1ad01042b9a11e1f65d610d1bd | 2024-07-07 19:04:43 (UTC) |
| 2024/07/07/[$LATEST]0b2a078edaa8433cbe993158d0f407f7 | 2024-07-07 17:52:28 (UTC) |
| 2024/07/07/[$LATEST]5e92fcdc3eef4b89a95e6e7de84163f9 | 2024-07-07 17:41:57 (UTC) |

6. Open a log stream to verify content

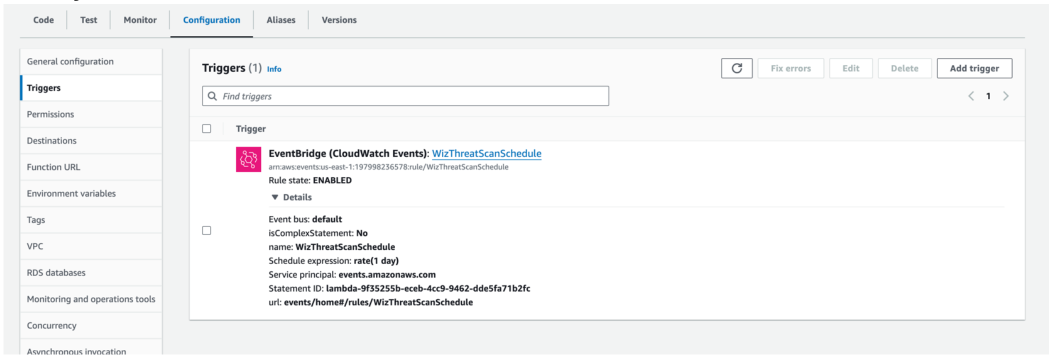


You can use the filter bar below to search for and match terms, phrases, or values in your log events. Learn more about filter patterns

Filter events - press enter to search

```
Timestamp                  Message
                           No older events at this moment. Retry
2024-07-08T02:52:37.733Z   INIT_START Runtime Version: python:3.12.v28 Runtime Version ARN: arn:aws:lambda:us-east-1::runtime:7776dee4c90fff4d4bf833b78d94e5b4148d14d044b867a14756f301ecfe1e28
2024-07-08T02:52:38.555Z   START RequestId: bb8b1d57-47b8-4767-9baa-897ed555a034 Version: $LATEST
2024-07-08T02:52:38.769Z   [INFO] 2024-07-08T02:52:38.769Z bb8b1d57-47b8-4767-9baa-897ed555a034 Found credentials in environment variables.
2024-07-08T02:52:43.009Z   debugMode True
2024-07-08T02:52:43.009Z   Getting token.
2024-07-08T02:52:44.020Z   checkThreatCenter initiated
2024-07-08T02:52:44.549Z   {'data': {'threatCenterItems': {'nodes': [{'id': 'regreSSHion', 'publishedAt': '2024-07-01T00:00:00Z', 'description': 'A critical unauthenticated RCE-as-root vulnerability was identified in OpenSSH server (sshd), assigned CVE-2024-6387 and dubbed regreSSHion…
2024-07-08T02:52:44.549Z   {'data': {'threatCenterItems': {'nodes': [{'id': 'regreSSHion', 'publishedAt': '2024-07-01T00:00:00Z', 'description': 'A critical unauthenticated RCE-as-root vulnerability was identified in OpenSSH server (sshd), assigned CVE-2024-6387 and dubbed regreSSHion…
2024-07-08T02:52:44.549Z   {
2024-07-08T02:52:44.549Z   "data": {
2024-07-08T02:52:44.549Z   "threatCenterItems": {
2024-07-08T02:52:44.549Z   "nodes": [
2024-07-08T02:52:44.549Z   {
2024-07-08T02:52:44.549Z   "alwaysShowFindingsCount": null,
2024-07-08T02:52:44.549Z   "description": "A critical unauthenticated RCE-as-root vulnerability was identified in OpenSSH server (sshd), assigned CVE-2024-6387 and dubbed regreSSHion. The vulnerability is a signal handler race condition that is exploitable in the default configuration…
2024-07-08T02:52:44.549Z   "docsUrl": "https://docs.wiz.io/wiz-docs/docs/wiz-adv-2024-090",
2024-07-08T02:52:44.549Z   "findingLinks": [
```

7. Verify schedule to ensure that Lambda function will run at desired interval



EventBridge (CloudWatch Events): WizThreatScanSchedule
arn:aws:events:us-east-1:197998236578:rule/WizThreatScanSchedule
Rule state: **ENABLED**

▼ Details

Event bus: **default**
isComplexStatement: **No**
name: **WizThreatScanSchedule**
Schedule expression: **rate(1 day)**
Service principal: **events.amazonaws.com**
Statement ID: **lambda-9f35255b-eceb-4cc9-9462-dde5fa71b2fc**
url: **events/home#/rules/WizThreatScanSchedule**