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

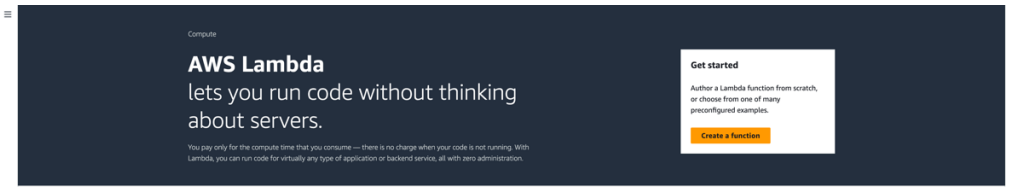
Implementation:
The script is implemented as an AWS Lambda function containing a single python file which is schedule to run on a desired repeating basis (e.g. daily). The Lambda function makes use of an SSM Parameter store to retrieve the following parameters. The function also updates the lastScanDate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

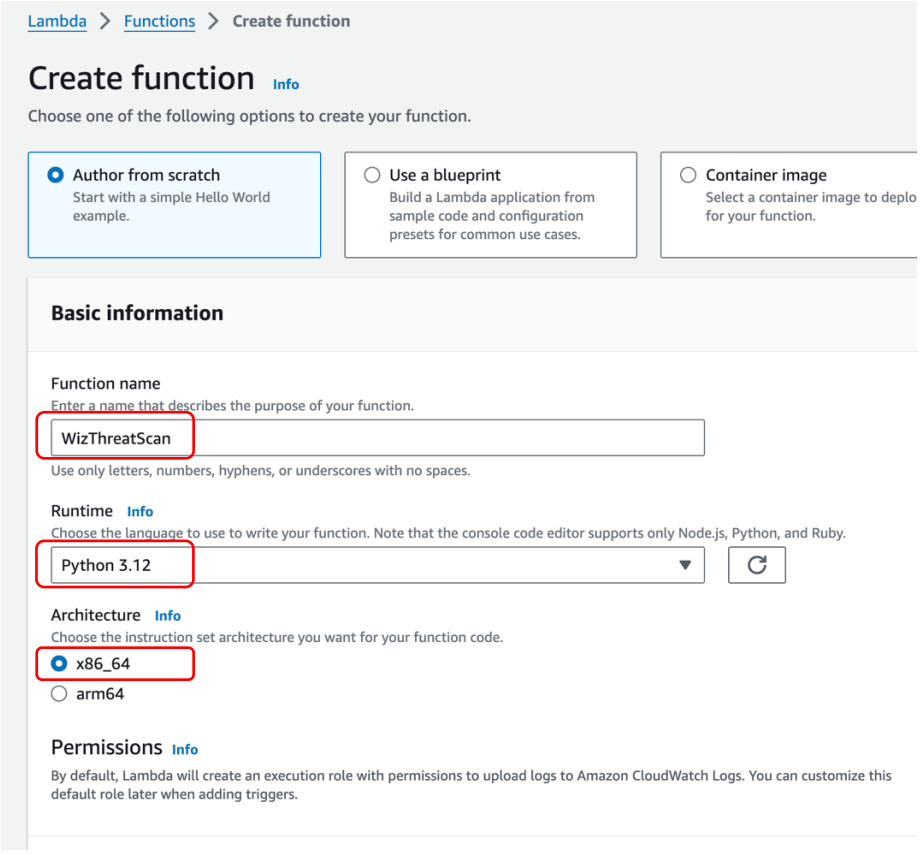
Installation and Setup:

AWS Lambda Setup:

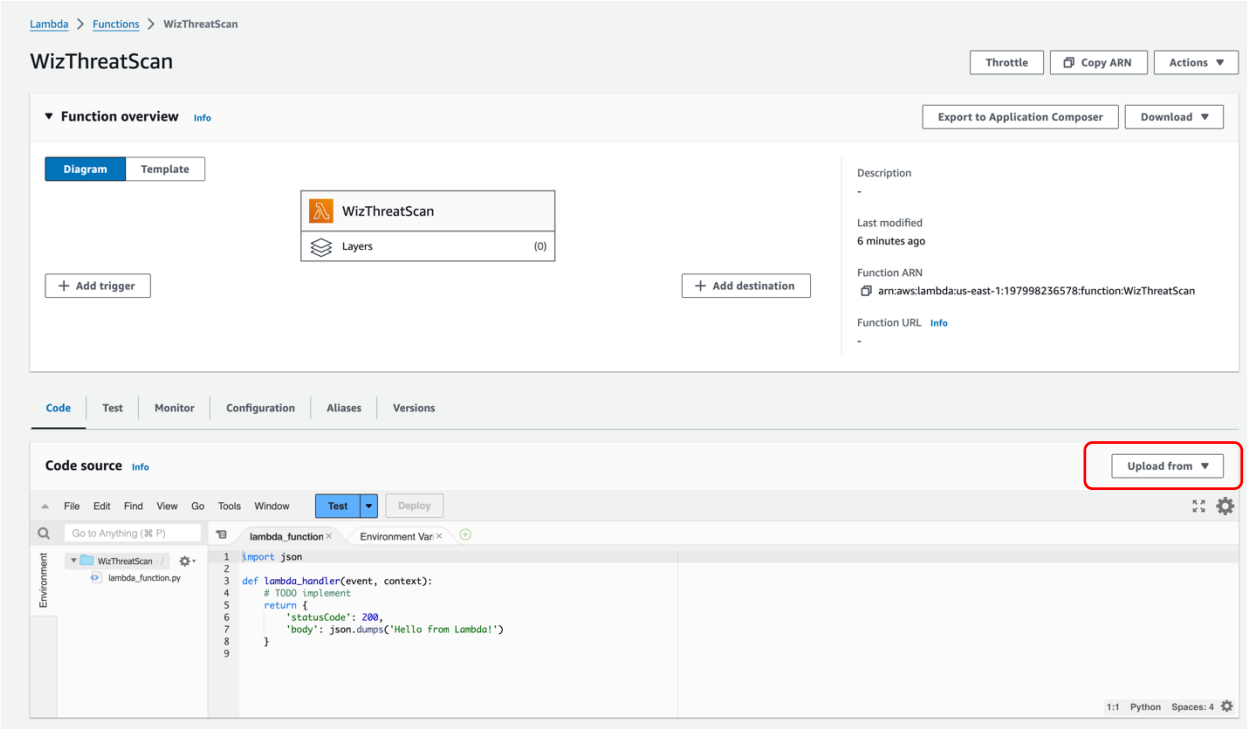
1. Download the WizThreatScan.zip and python_package.zip files (git clone https://github.com/jsing3r/WizThreatScan.git)
2. Login to AWS console with Admin rights (create roles, assign role, create SSM, create Lambda function, etc.)
3. Set your desired region (e.g. us-east-1)
4. Create a new Lambda function



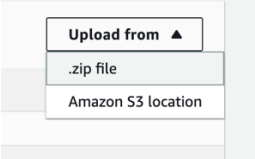
5. Use the information in the following image to create your Lambda function:



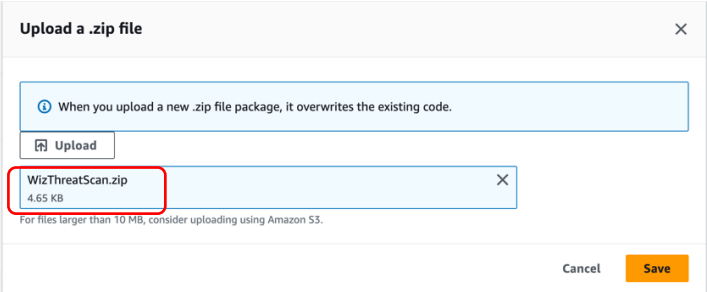
6. You should have a default Lambda function created as shown below:



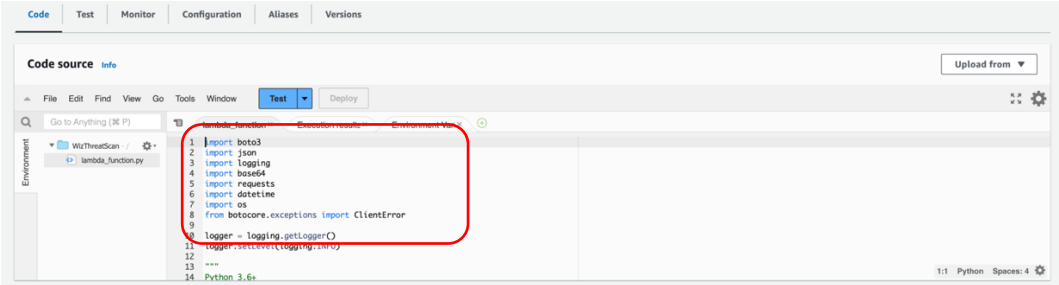
7. Select Upload from option and select .zip file



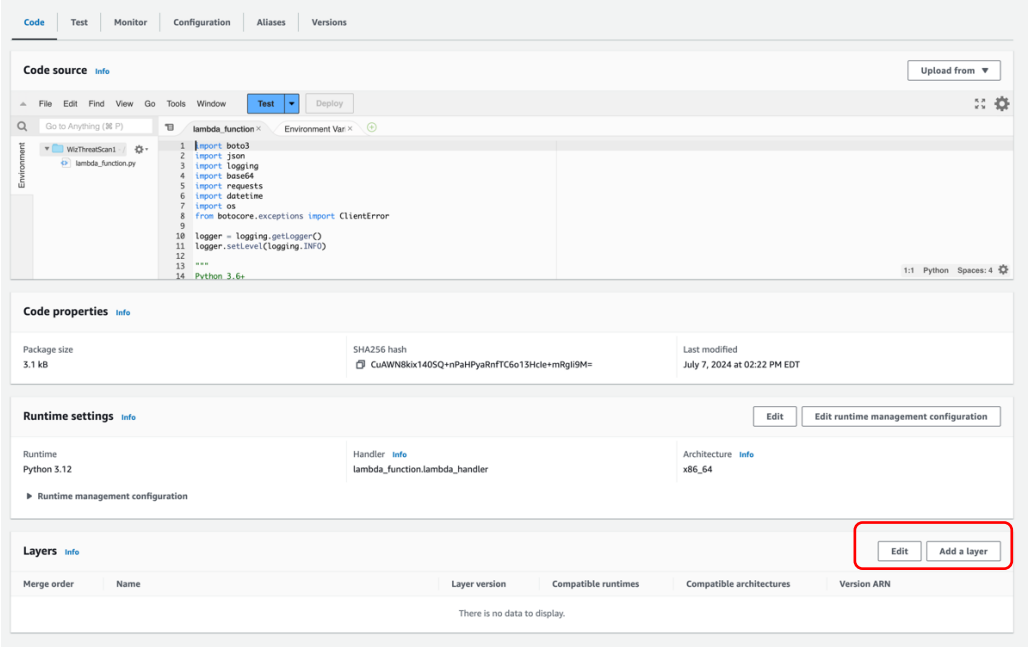
8. Select the WizThreatScan.zip file saved earlier to update the Lambda code



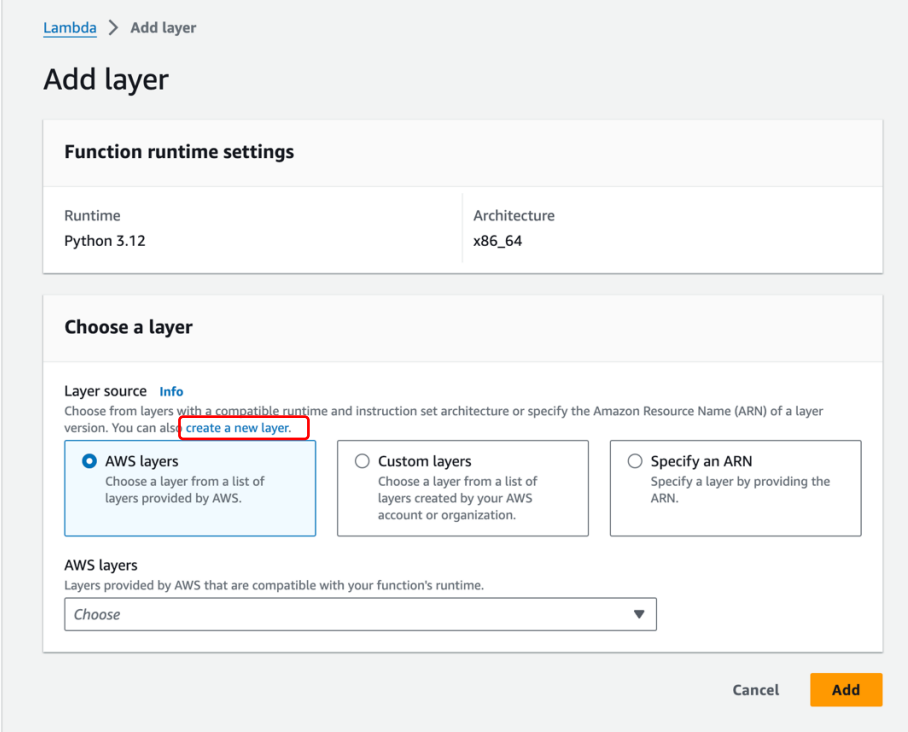
9. Confirm that the lambda_function.py has been updated as shown below



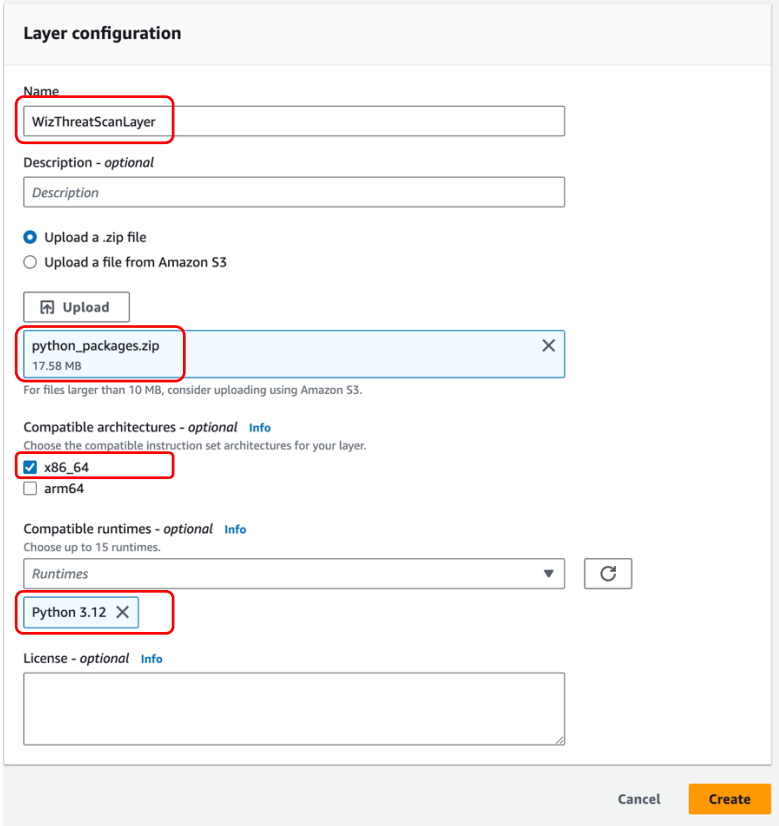
10. Scroll down to the Layers section on the Lambda page



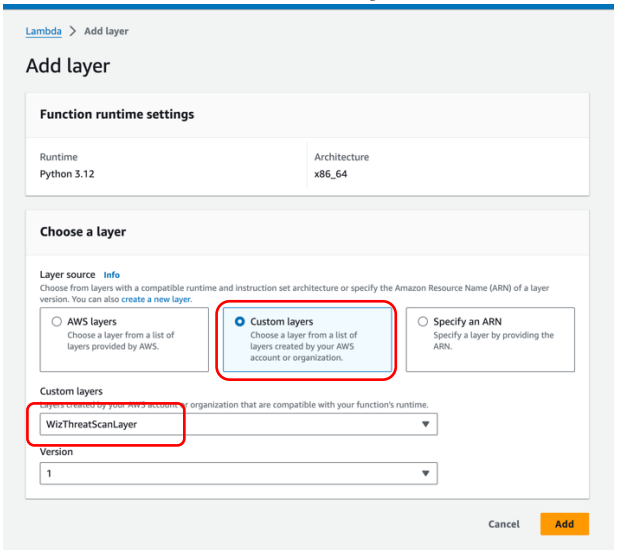
11. Select Add a layer, and then select **create a new layer**



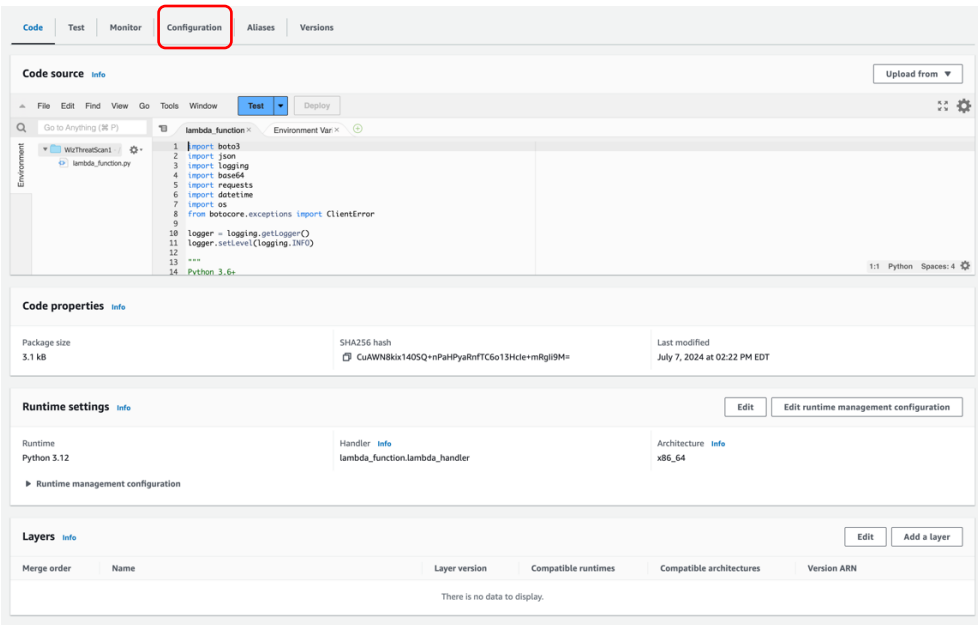
12. Create a new layer called WizThreatScanLayer and upload the python_packages.zip file to the layer. Also select the compatible architecture and runtimes as shown



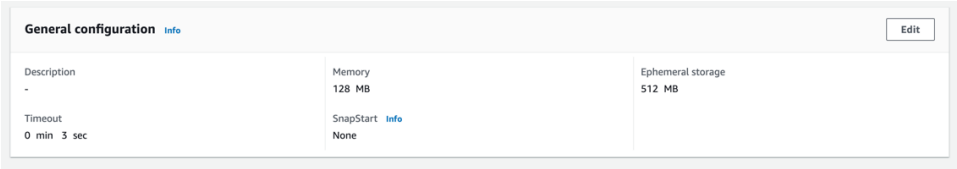
13. Return to the Add a layer dialog, select Custom layers, then select the newly created **WizThreatScanLayer** and Version 1 in the pulldown list:



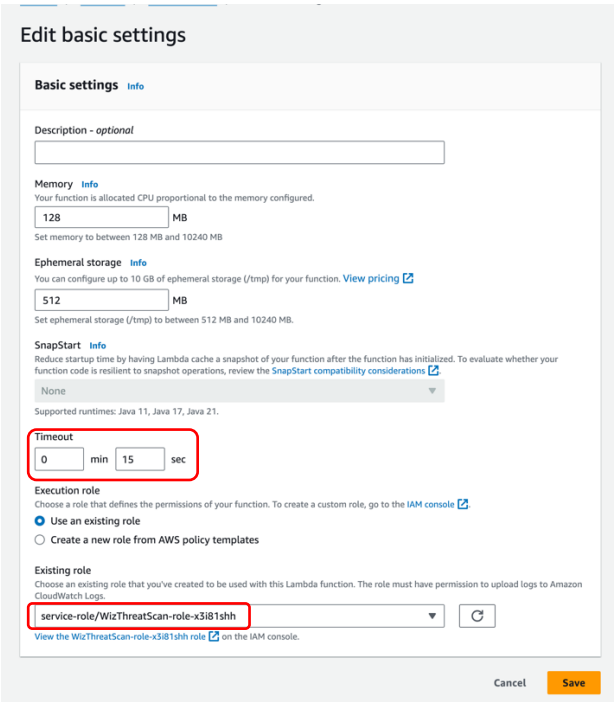
14. Navigate to the **Configuration** tab



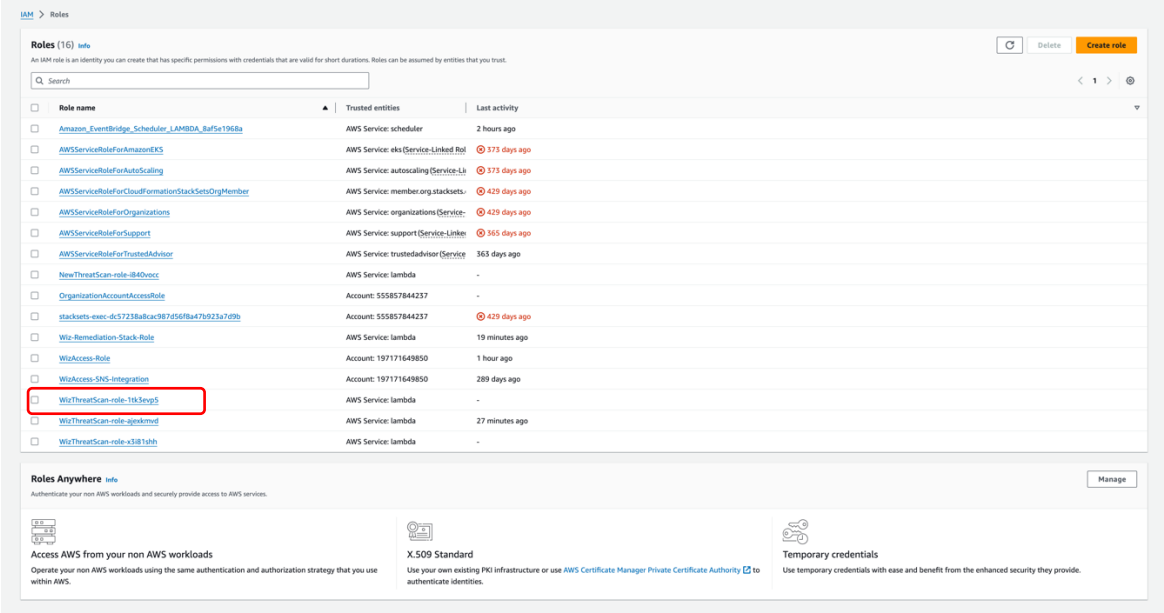
15. Under General configuration, select Edit



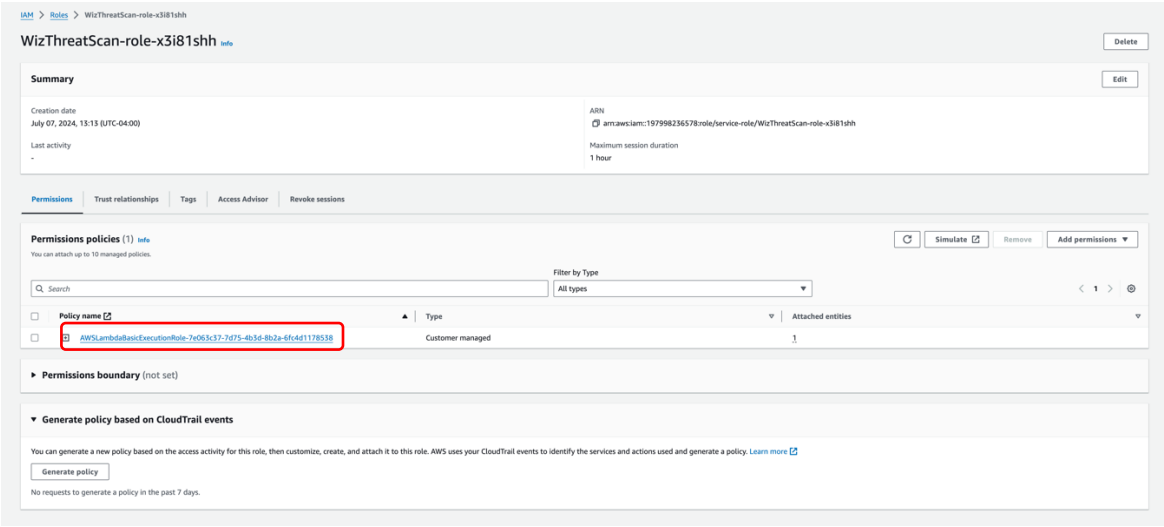
16. Set the Timeout value to 15 sec instead of the default of 3 sec. Also make note of role that will be used to execute the Lambda function



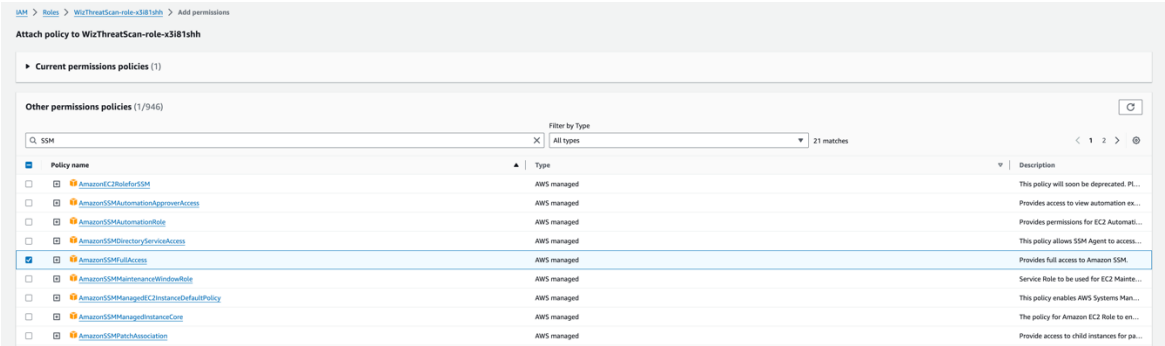
17. Navigate to IAM in the AWS console, and select Roles. Locate the **WizThreatScan** role identified above:



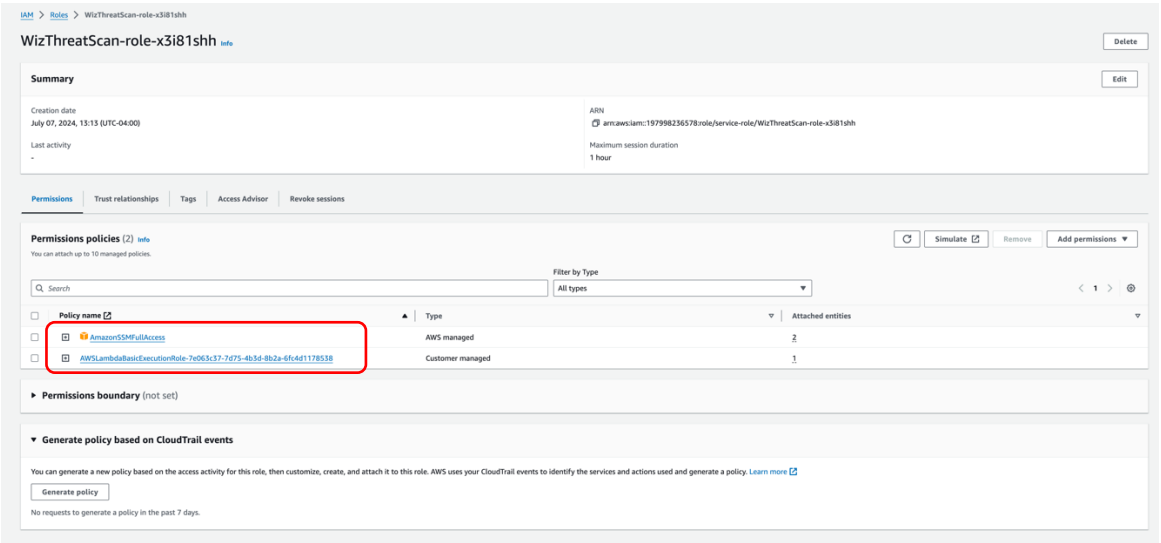
18. Select the **WizThreatScan** role and confirm the permission policies are like that shown below



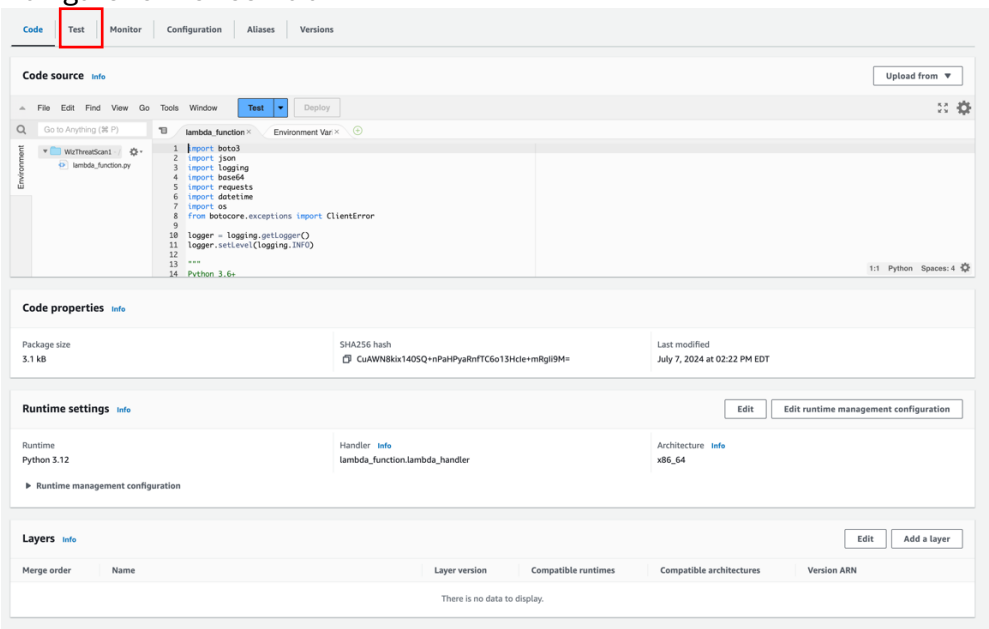
19. Select Add permissions and Attach policy, then select **AmazonSSMFullAccess** policy



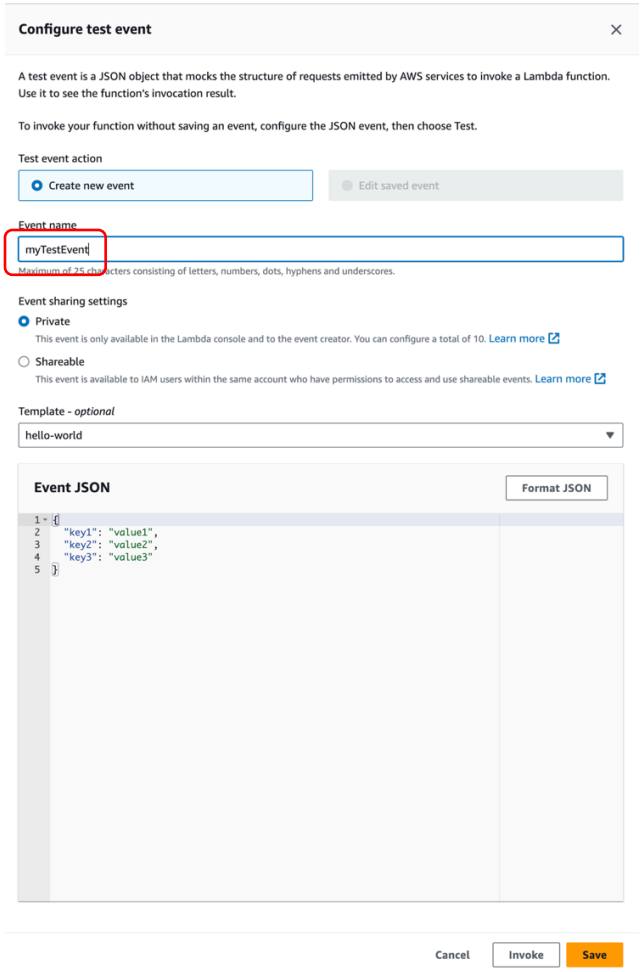
20. Confirm that the WizThreatScan role now has the **AmazonSSMFullAccess** policy attached.



21. Navigate to the Test tab



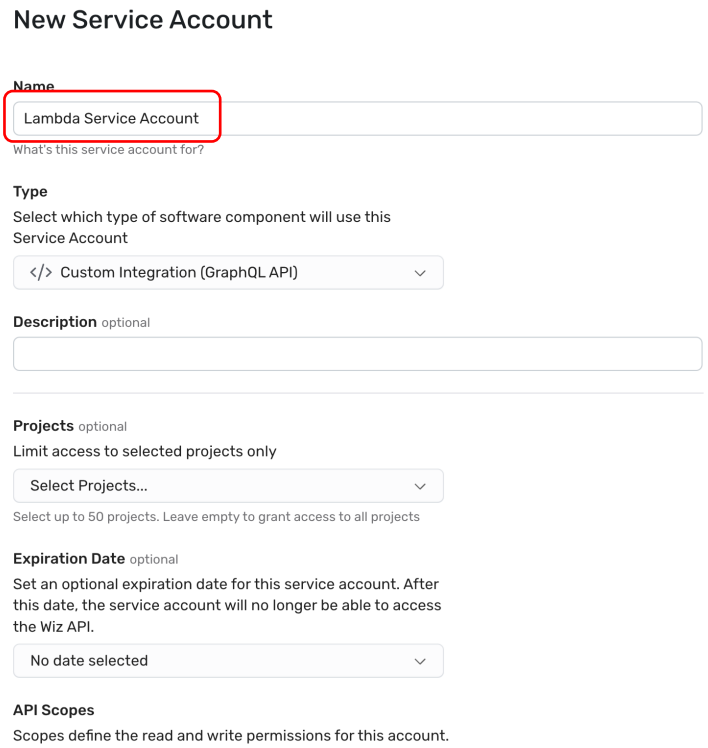
22. Create a new test event called **myTestEvent** as shown. This test data will not be used but it allows the Test button to be used to test our function.



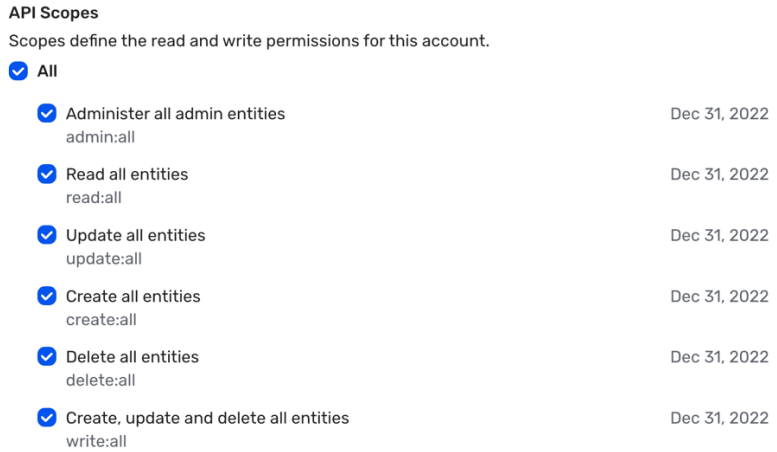
Wiz UI Setup:

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

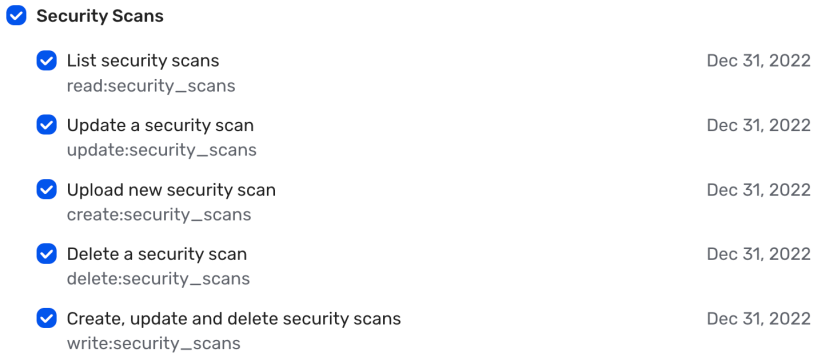
2. Create a new Service Account of type Custom Integration (GraphQL API)



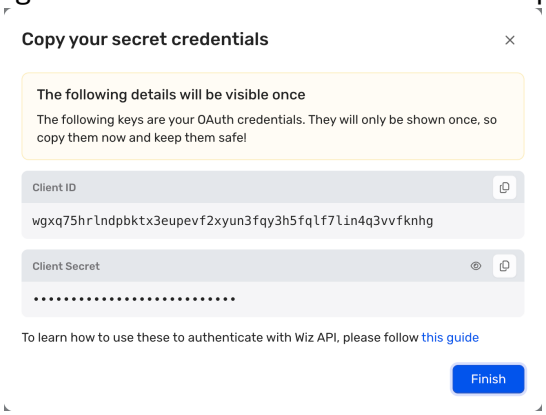
3. Select all API Scopes



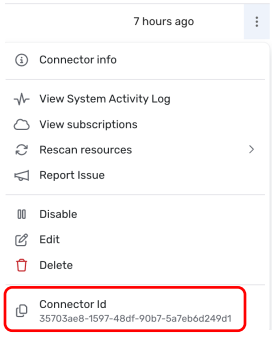
4. Select all Security Scan Scopes



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

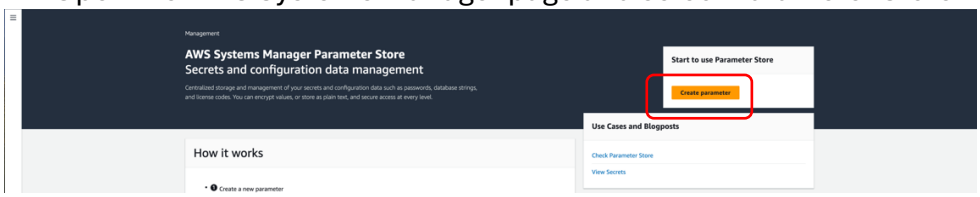


6. Open the Wiz Settings / Deployments page and identify the connectors that should be rescanned for new threats. Click the menu at the end of the connector line to find the connector ID. These IDs will be entered as comma-separated values in the SSM parameter list.



AWS SSM Setup:

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

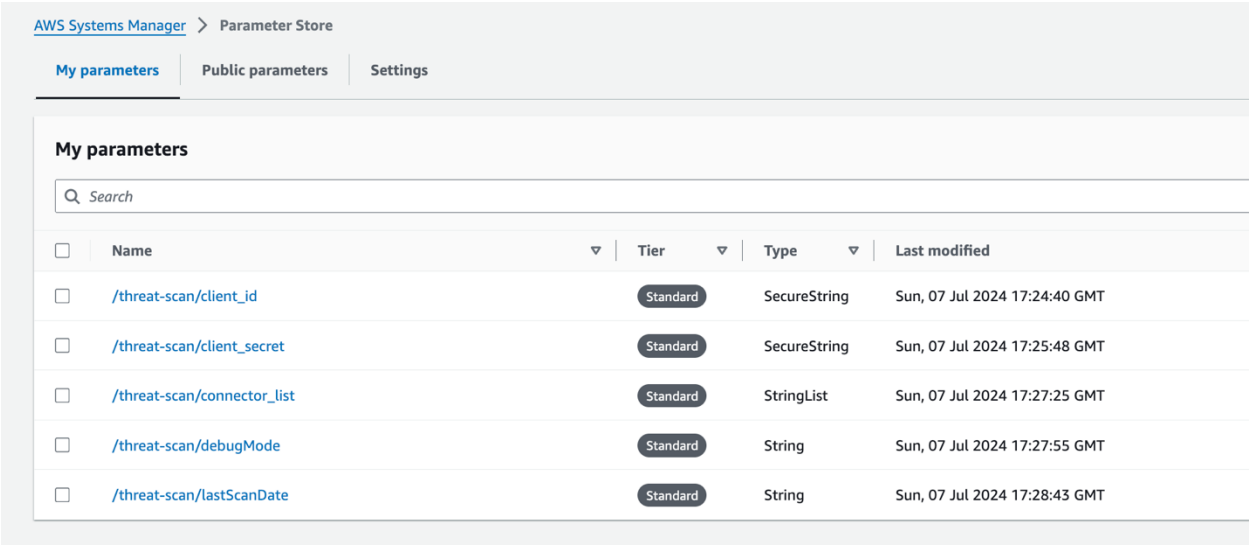


2. Create the following parameters :

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

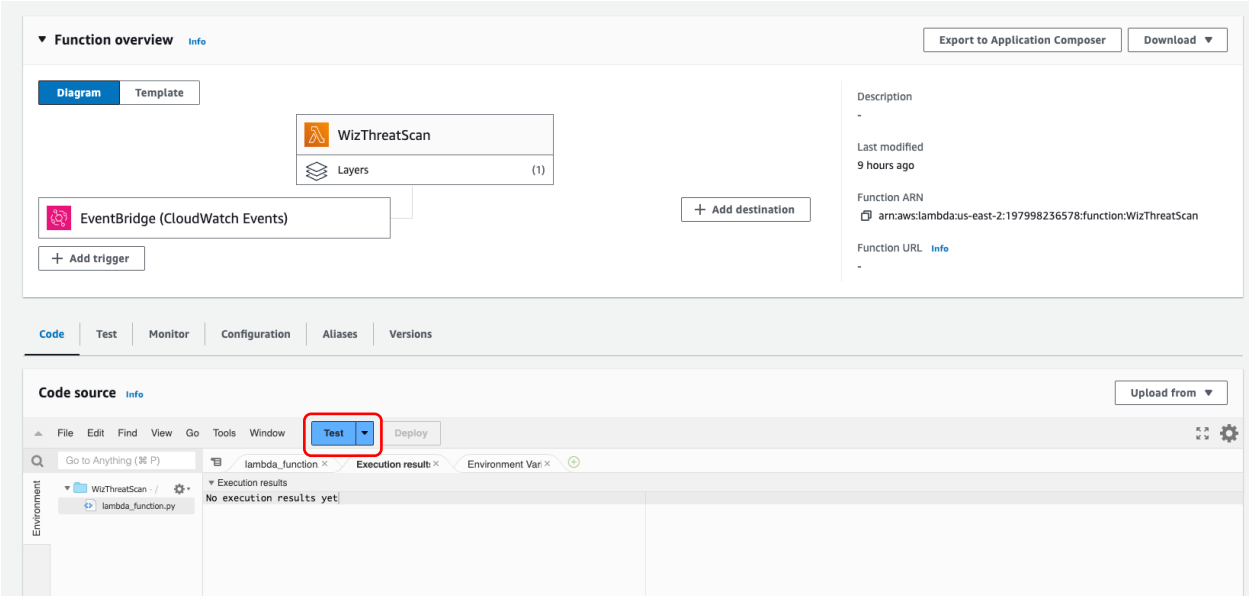
/threat-scan/debugMode	String	True or False (Set to True for detailed debugging information)
/threat-scan/lastScanDate	String	Must contain some initial value which will be rewritten with the current date if invalid. A good starting example is NEVER.

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

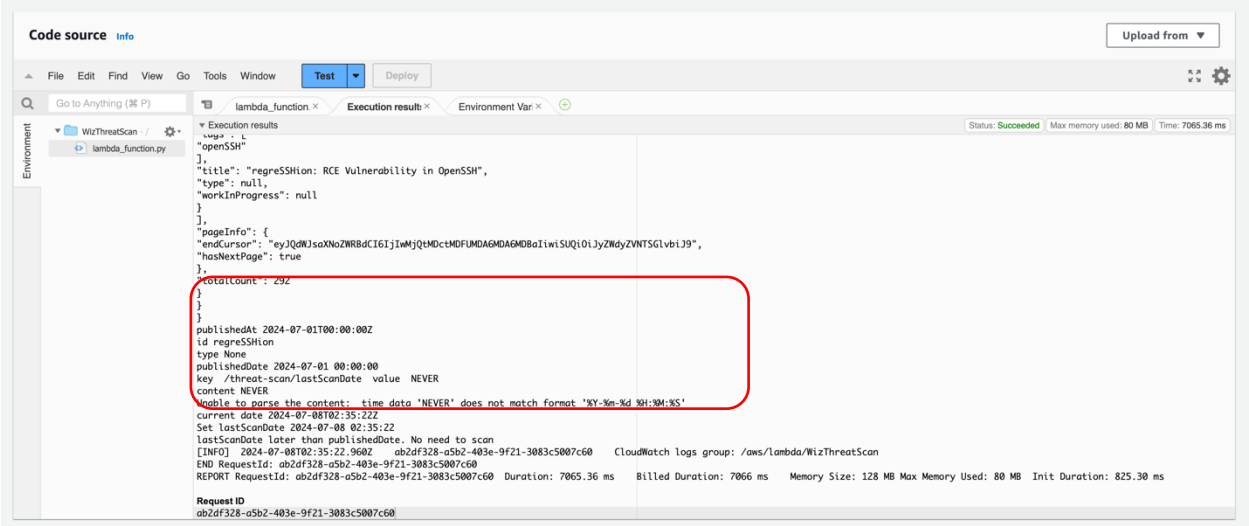


Testing the script:

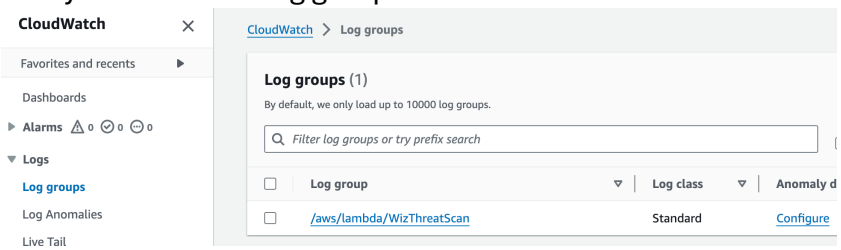
1. Configure all setup elements in AWS and Wiz as described above
2. Select Test button



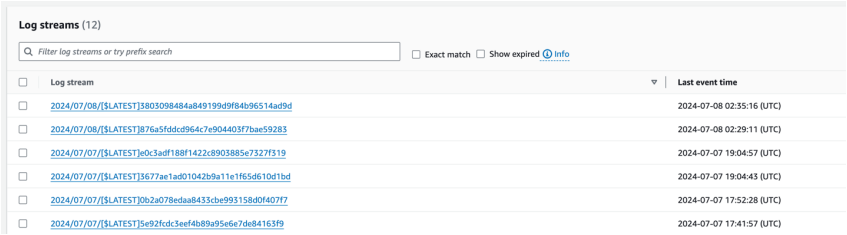
3. Observe output in execution log window. For first invocation, lastScanDate (set to NEVER in this example) is overwritten with the current date. For every following run, the lastScanDate will be compared with the date of the most recent threat center notice, and a rescan will be triggered as required.



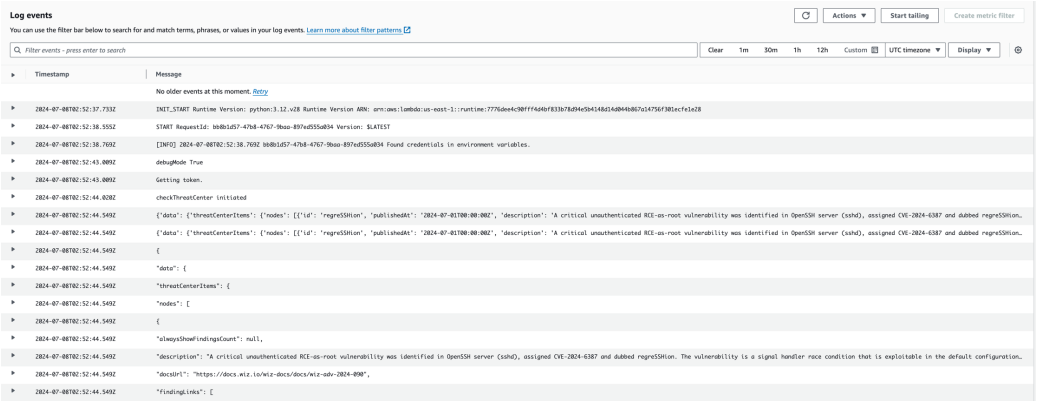
4. Verify CloudWatch log group for the Lambda function



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



6. Open a log stream to verify content



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