Lambda function AWS2ZabbixEC2Maintenance Installation Guide

Client logo

Tooling & Automation

Partner logo

Contents

[1. Introduction 3](#_Toc83640935)

[2. Solution description 4](#_Toc83640936)

[3. General information 4](#_Toc83640937)

[4. Used services 5](#_Toc83640938)

[5. Call method 5](#_Toc83640939)

[6. Return 6](#_Toc83640940)

[7. Deploy process 6](#_Toc83640941)

[7.1. Setup communication to AWS account 6](#_Toc83640942)

[7.2. Adjusts Terraform variables values 6](#_Toc83640943)

[7.3. Apply the configuration 8](#_Toc83640944)

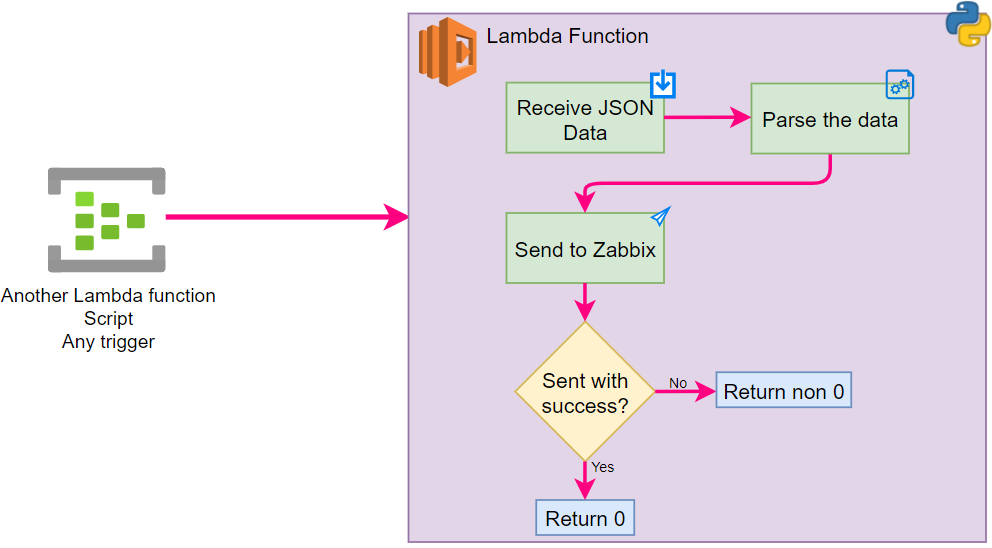
[7.4. Validate the communication 8](#_Toc83640945)

# Introduction

This document explains how to install lambda function AWS2ZabbixEC2Maintenance.

# Solution description

Deploy all components to inform Zabbix that the EC2 instance is temporarily stopped and needs to be put into maintenance mode.



The terraform code is available in aws2zabbixec2maintenance.zip archive.

# General information

This function will notify the Zabbix when the a standalone EC2 is shutdown in order to put the host into maintenance mode..

How all communication must be done safely, the Lambda function is deployed in some VPC to be able to communicate through a VPN, VPN peering or other safe way.

# Used services

* Lambda function
* IAM Role and Policy
* Custom security groups

# Call method

This lambda function can be called by several ways, below we have a example in Python:

import json

from boto3 import client as boto3\_client

def lambda\_handler(event, context):

lambda\_client = boto3\_client('lambda')

event = {"body": { "instanceid": "abc123z", "maintenance": "1" }}

res = lambda\_client.invoke(FunctionName='recebe',

InvocationType='RequestResponse',

Payload=json.dumps(event))

ret = res['Payload'].read().decode()

if ret == "0":

print("OK")

else:

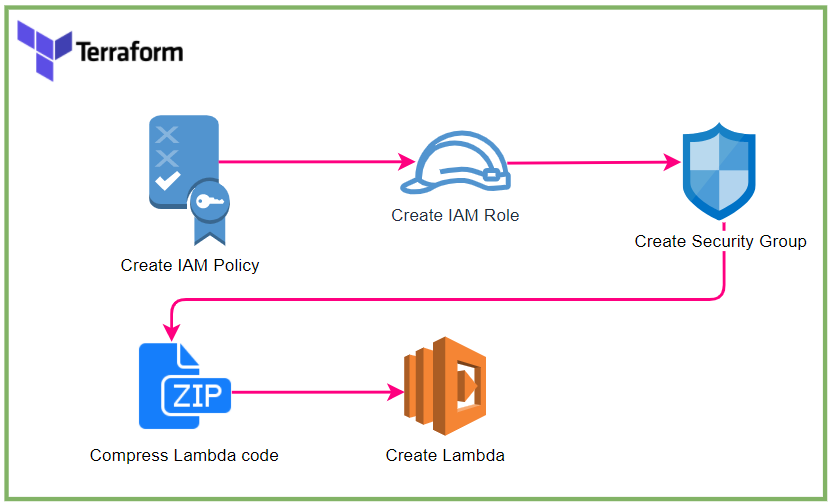
print("erro")

if the sender is another Lambda function ensure that the execution rule assigned has at least "lambda:InvokeFunction" permission.

# Return

If the data was successfully sent to Zabbix will be returned 0, else, a non 0 value.

# Deploy process



## Setup communication to AWS account

Configure the credentials to communicate to AWS.

This step depending of the customer CI/CD solution, on Azure DevOps this step can be done adding a Service Connection in the project.

## Adjusts Terraform variables values

All variables must be changed on Terraform variable file (terraform.tfvars).

They were divided into groups:

**General** (General data that may be used for others scripts)

|  |  |  |  |
| --- | --- | --- | --- |
| Variable | Type | Description | Default |
| region | string | AWS region | eu-west-3 |
| env | string | Environment of the solution | prd |
| app | string | Application name | Inetum Monitoring |

**lambdaEC2Maintenance** (These variables are specific for AWS Lambda function)

| Variable | Type | Description | Default |
| --- | --- | --- | --- |
| name | string | Name of Lambda function | Inetum\_AWS2ZabbixEC2Maintenance |
| description | string | Description of Lambda function | Function to notify Zabbix about Manage maintenance window |
| vpcId | string | VPC ID to deploy the Lambda function | NULL |
| subnetIds | array | Subnets ID to deploy the Lambda function | NULL |
| logRetention | number | Retention of CloudWatch Lambda execution (days) | 30 |
| timeout | number | Timeout to execute Lambda function | 10 |

**zabbix (Information about Zabbix connection to be used in Python script on Lambda function)**

| **Variable** | **Type** | **Description** | **Default** |
| --- | --- | --- | --- |
| host | string | The IP Address or FQDN of the Zabbix Proxy (Proxy must be reacheable from the account automation) | NULL |
| port | Integer | TCP port used by Zabbix proxy to listen incoming alerts | 10052 |
| trapItem | string | Name of item on Zabbix configured to receive these alerts | server\_down |
| timeout | number | timeout to connect to Zabbix server | 5 |

By default, the TF code will apply some TAGs, if necessary you can change the alarm\_tags object in variables.tf file

Will be used the instanceID as clientName

Adjust the timeouts if necessary

## Apply the configuration

This step depending the customer CI/CD solution.

In Azure DevOps you might need to run a Pipeline and/or a Release.

In terraform CLI you might need to run the command *terraform apply*.

## Validate the communication

Get an instance that exists on Zabbix and shutdown it, after that ask to Inetum Tooling team to check in Zabbix f the server is in maintenance mode.

### Révisions

|  |  |  |
| --- | --- | --- |
| Version | Date | Objet |
| 1.0 | 27/09/2021 | Création of document |
|  |  |  |
|  |  |  |

### Visas

|  |  |  |  |
| --- | --- | --- | --- |
|  | Responsable | Date | Visas |
| Rédaction | Sophie Dupont | 27/09/2021 | SDU |
| Vérification |  |  |  |
| Approbation |  |  |  |

### Diffusion

|  |  |
| --- | --- |
| Entreprise | Destinataire |
| INETUM | Skill Center IS |
|  |  |
|  |  |