VERCARA

Configuring UltraDNS Realtime Push Notifications - Webhook in Lambda

Customer Guide

Version 1.0 (July 2023)

Find more information at:

vercara.com

Call USA: +1 (844) 929 - 0808
Call EMEA: +44 808 175 1189

AWS API Gateway and Lambda functions to handle UltraDNS Push Notifications

When UltraDNS calls webhook endpoints with a notification it may be desirable to re-format the message or process the message into other systems. A very simple way to achieve this notification proxy is to use AWS. Setting up an AWS API Gateway to forward the notifications to a Lambda function gives the ability to handle the notifications in a way that better suits the payload requirements of the ultimate destination.

The following process uses the AWS console to build a handler for UltraDNS webhook calls.

Setup Webhooks with an AWS Lambda Function and API Gateway

1. **Create the Lambda function.**
    a) Select the **Author from scratch** option.
    b) Provide a **Function Name** that will help you remember the purpose of the function.
        I. For example: `ultradns-webhook`
    c) Select the **Runtime** option.
        I. Use the latest supported python version. In the following example, we are using **Python 3.10**.
    d) Select the (instruction set) **Architecture** value.
        I. **x86_64** is recommended.
    e) **Advanced Settings** configuration:
        I. **Enable Tags** – This is not required, but it will come in handy for your infosec and administrative watchers.
    f) Create function

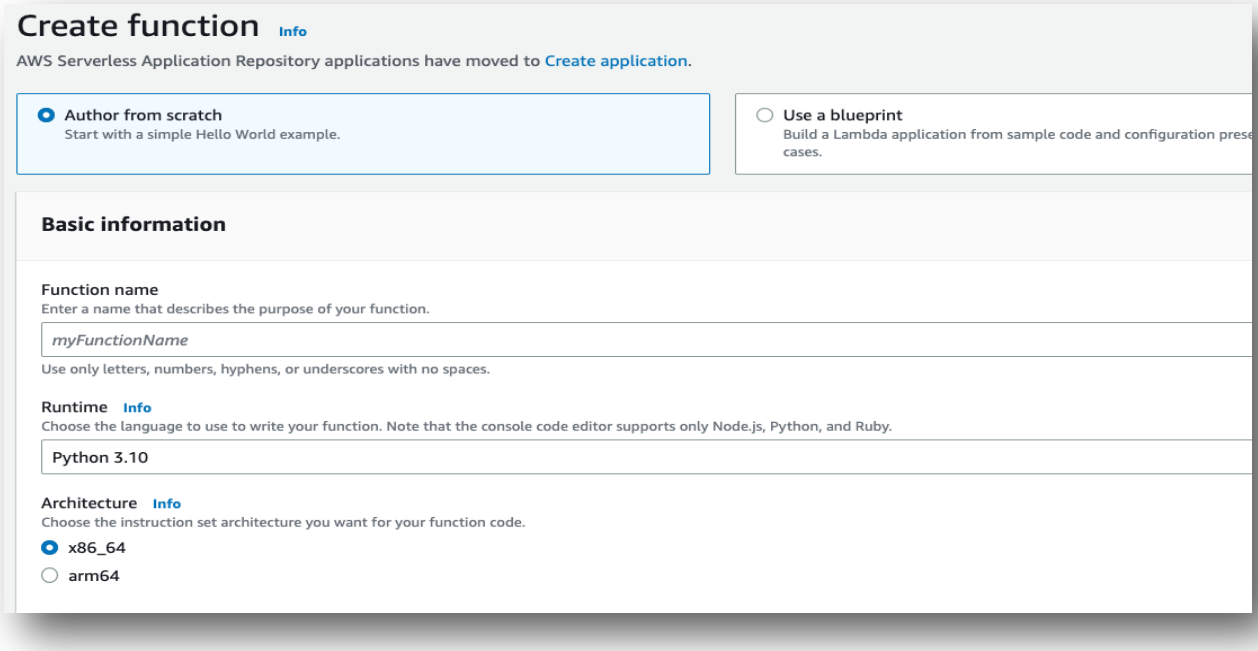

Figure 1 Lambda - Create Function Configuration

g) Write the code

I. An example is provided in github to parse, format, and send notifications to Slack endpoint.

https://github.com/ultradns/samples/blob/master/webhook/lambda-slack.py

2. **Create the API Gateway**

a) Select the **REST API** option - AWS allows you to configure Source IP access controls for REST API gateways. General HTTP gateways do not have this feature.

b) From the **Choose the Protocol** section, select **REST**.

c) In the Create **New API section**, select **New API**.

d) **Settings Configuration**

i. Provide the **API Name**. (i.e., 'ultradns`)

ii. From the **Set Endpoint** dropdown menu, select **Regional**.

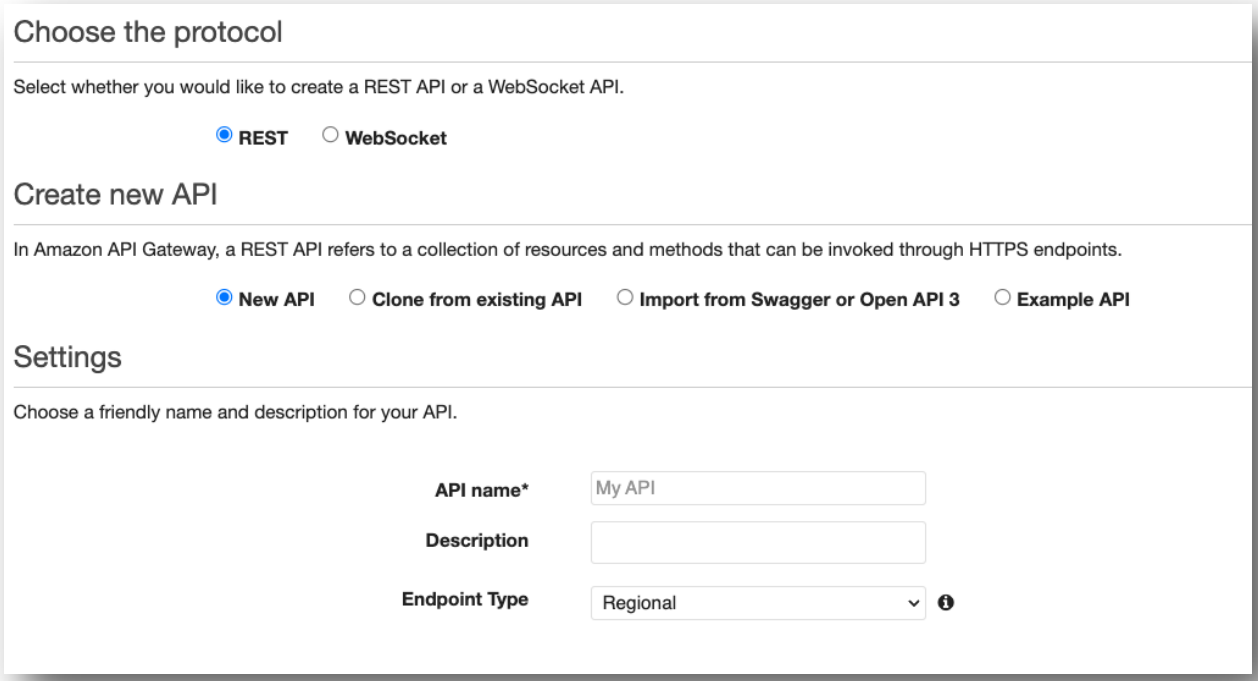

Figure 2 Lambda - REST API Gateway Configuration

3. Once the gateway is created you will need to **create a resource**.
    a) Select your new API gateway that will be configured.
    b) Click on **Resources**.
    c) Select **Actions** and then **Create Method**.
    d) Choose **POST**.
    e) Set the **Integration Type** to **Lambda Function**.
    f) Select the **Lambda Region.**
        I. This is the same AWS region where the lambda function was created.
    g) Set the **Lambda Function** to the same name you provided in the initial Lambda setup (*Create the Lambda function.*).
        I. In our example, it was 'ultradns-webhook'.
    h) Configure a **Resource Policy** to restrict who can call this API. Set the Allow List to include the IP addresses given in the UltraDNS documentation for Realtime Push Notifications.
        I. The Resource Policy can be left "wide" open during development, and then configured afterwards for greater restrictions as necessary.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": "*",
      "Action": "execute-api:Invoke",
      "Resource": "arn:aws:execute-api:us-east-
1:XXXXXXXXXX:XXXXXXX/*",
      "Condition": {
        "IpAddress": {
           "aws:SourceIp": [
              "52.87.134.132",
              "52.201.155.120",
              "52.201.103.62",
              "52.201.155.234",
              "52.10.123.90",
              "52.10.63.3",
              "52.39.68.132",
              "52.214.64.69"
           ]
        }
      }
    }
  ]
}
```

4. **Deploy the API**

   a) Under the **Select Stages** option, select **Create**.

   b) Provide a deployment stage name.

      I. For example, we used `webhook`.

      II. Copy the **Invoke URL** as this will need to be added to the UltraDNS Managed Services Portal configuration.

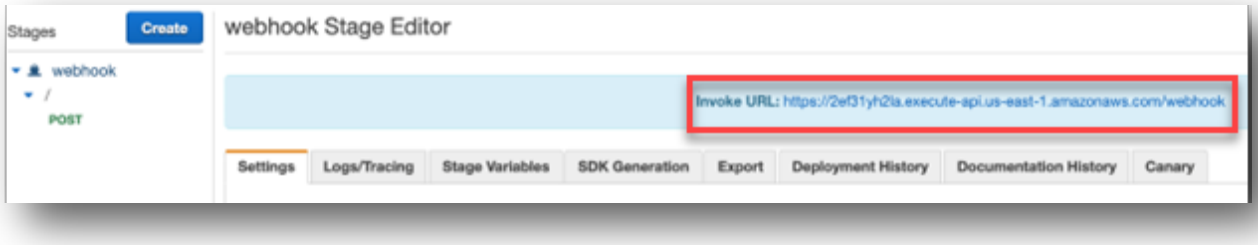

Figure 3 Lambda - Webhook Stage Editor

   c) Select **Resources**, then **Actions**, and then **Deploy API**.

UltraDNS Managed Services Portal Configuration

Configure real-time push notifications in UltraDNS using the UltraDNS Managed Services Portal.

NOTE: Administrative user privileges are required to configure the Realtime Push Notifications on the UltraDNS portal.

1. Once you are logged into the portal, navigate to **Accounts** on the left-hand navigation menu.
2. Select the **Account Name**.
3. Select **Notification Settings** from the list of available features.

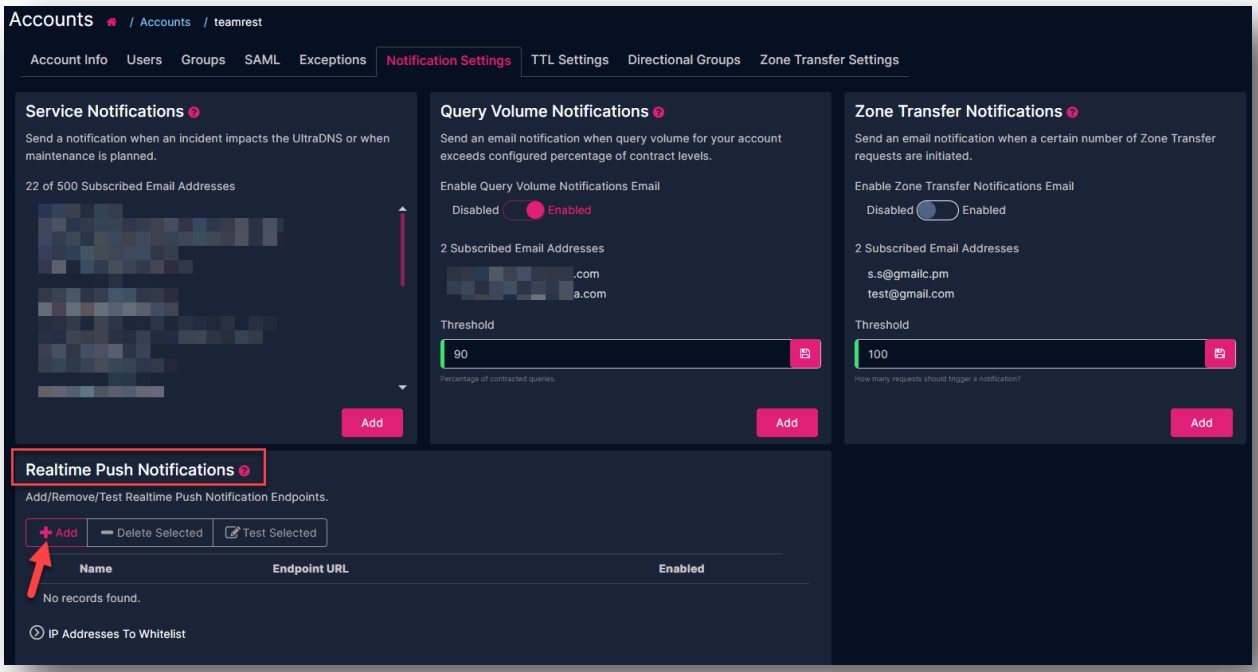

Figure 4 UltraDNS Portal- Add Realtime Push Notification

4. In the **Realtime Push Notifications** section, click the **Add** button to configure a new endpoint.

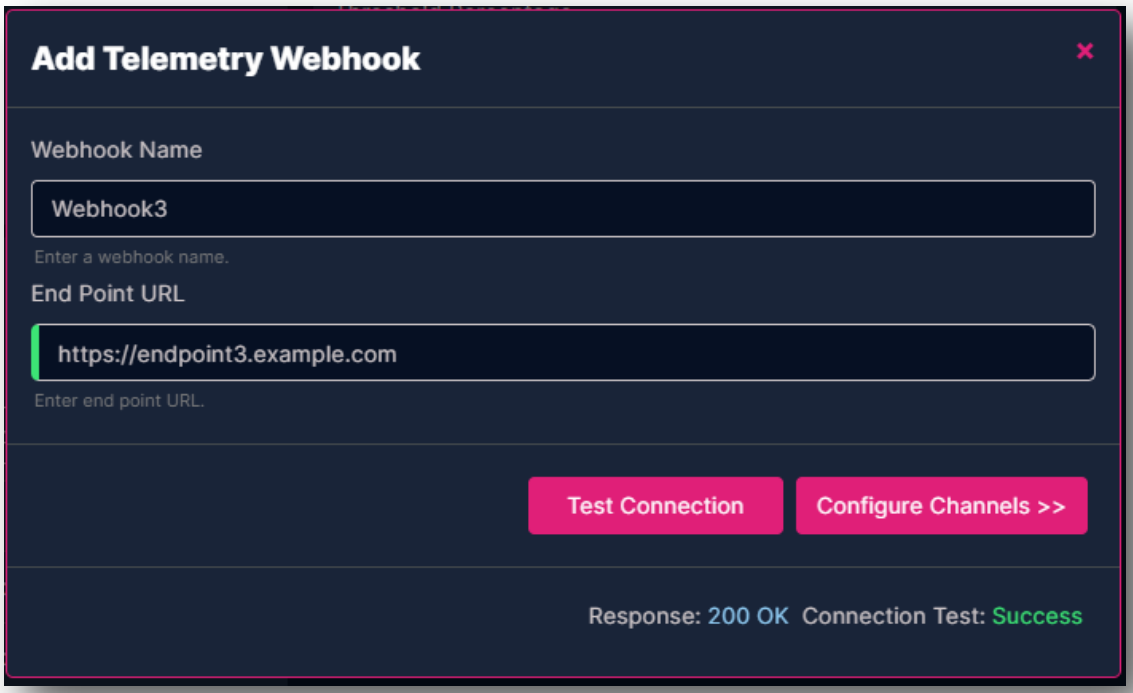

Figure 5 UltraDNS Portal - Configure Webhook

5. Create a **Webhook Name**, and then enter the **Endpoint URL**.
    a) The Endpoint URL must begin with https.

> The Endpoint URL will be the Invoke URL from the configuration of the API Gateway configured above. For example:
>
> `https://XXXXXXXX.execute-api.us-east-1.amazonaws.com/webhook`

6. Click the **Test Connection** button. You need to have a successful connection test before proceeding to the channel configuration.
7. Click **Configure Channels** once your Connection Test displays *Success*.
8. Select the **Channels and Events** you want to receive notifications for from the list of available options.

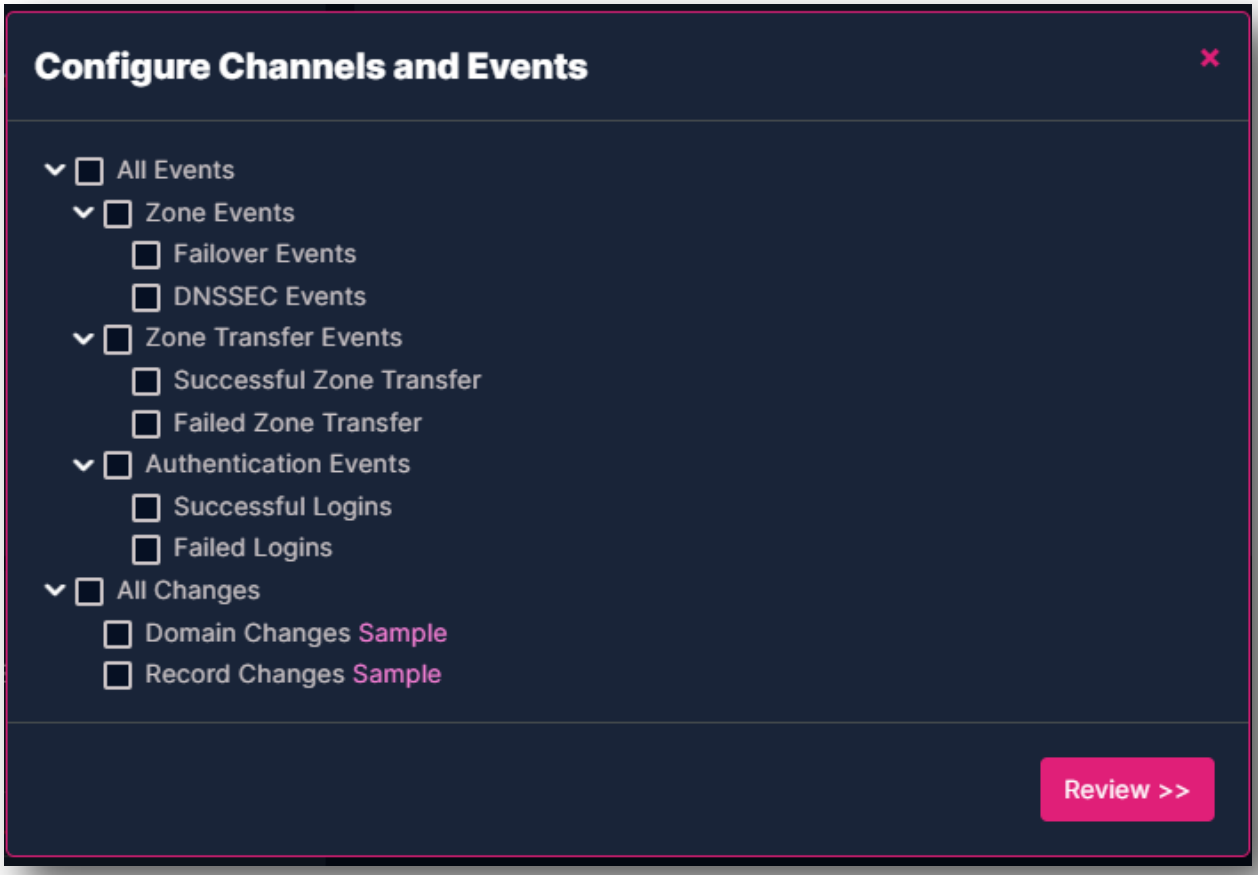

Figure 6 UltraDNS Portal - Configure Channels and Events

9. **Review** and **Save** your webhook configuration.
10. Make sure your new configuration is set to **Enabled** to begin receiving notifications.

The UltraDNS service is now configured to send notifications to your AWS Lambda function. At any time, you can test the connection or turn the notifications off through the UltraDNS Portal interface.