ExtraHop 6.0

Open Data Stream walkthrough: Sending ExtraHop metric data to AWS CloudWatch

Published: 2017-11-15

The ExtraHop system provides several tools for viewing and monitoring metrics about your network data. However, you might want to to store or analyze metric data with a remote, third-party tool, such as Splunk, MongoDB, or Amazon Web Services (AWS). The Open Data Stream (ODS) feature enables you to configure a connection to a third-party tool through which you can send specified metric data.

In this walkthrough, you will configure an ODS target for Amazon CloudWatch, write a trigger that specifies which HTTP metrics to send, and initiate the transmission of data to the target.

Prerequisites

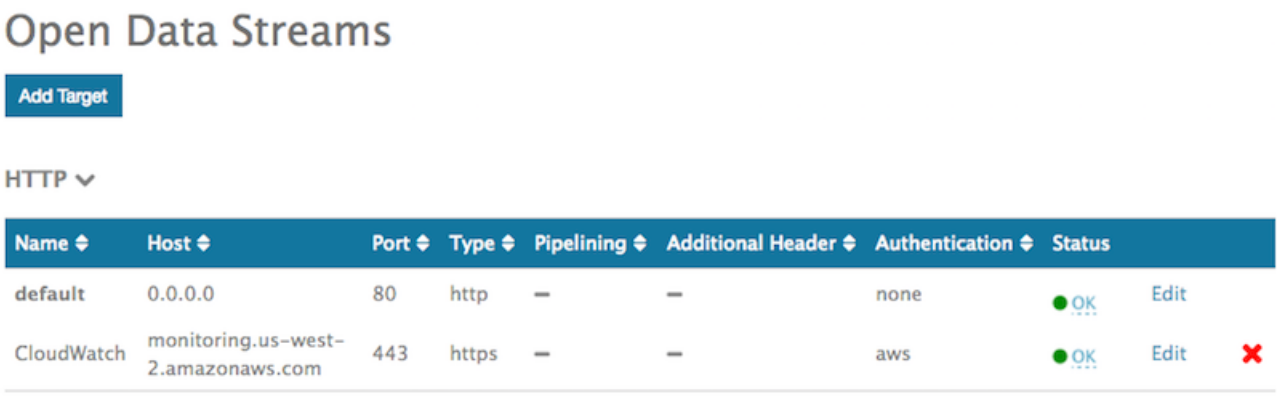
- You must have access to an ExtraHop Discover appliance with a user account that has full system privileges.
- Your ExtraHop appliance must have network data with web server traffic.
- You must have an Amazon Web Services account and familiarity with the CloudWatch service.
- Familiarize yourself with the concepts in this walkthrough by reading the Open Data Streams section in the ExtraHop Admin UI Guide and the Get started with triggers section in the ExtraHop Web UI Guide.
- Familiarize yourself with the processes of creating triggers by completing the Trigger Walkthrough.

Configure an ODS target

In the following steps, you will configure the host, port, and authentication method for an HTTP ODS target.

1. Log into the ExtraHop Discover appliance that you want to send data from with an account that has full system privileges.
2. Click the System Settings icon, and then click **Administration**.
3. From the System Configuration section, click **Open Data Streams**.
4. Click **Add Target**.
5. Select **HTTP** from the Target Type drop-down list.
6. In the Name field, type `CloudWatch`.
7. In the Host field, type the IP address or hostname of the Amazon web server you want to send data to.
8. In the Port field, type 443 for the port number you want to send data through.
9. In the Type field, select **HTTPS** as the transfer protocol you want to send data through.
10. In the Authentication field, select **Amazon AWS**.
11. In the Access Key ID field, type the access key for your AWS account.
12. In the Secret Key field, type the secret key for your AWS account.
13. In the Service field, type the entry point for the CloudWatch service, such as monitoring.
14. In the Region field, type the region for the CloudWatch service, such as us-west-2.
15. In the Method field, select **POST** as the REST method the trigger will call when sending data.
16. Click **Save**.

The target is added to the HTTP table on the Open Data Stream page, similar to the following figure:



Test the ODS configuration

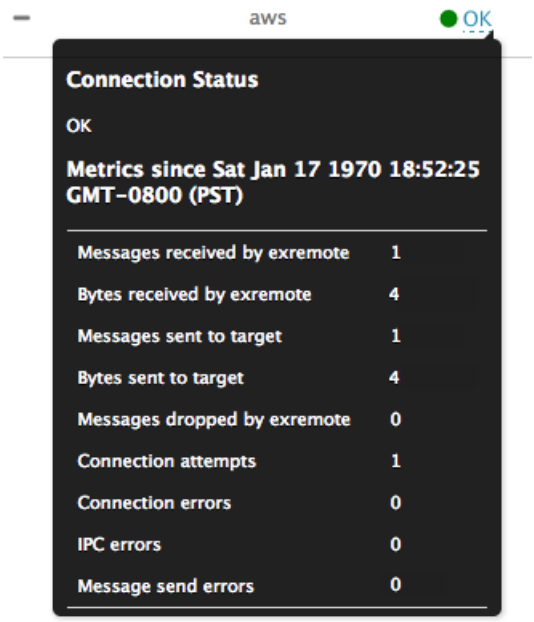
In the following steps, you will write an HTTP REST request to test the transmission of data from the ExtraHop Discover appliance to the AWS account.

As configured in the previous section, the test request applies the POST method.

1. In the HTTP table, click **Edit** to open the CloudWatch target.
2. In the Options field, copy and paste the following HTTP REST request code to send a metric called "Test" with a value of 4 bytes to the CloudWatch service:

```
{
    "path": "/",
        "payload":
 "Action=PutMetricData&Version=2010-08-01&Namespace=test&MetricData.member.1.MetricN
            "headers": {
                "Content-Type": [
                    "application/x-www-form-urlencoded"
                ]
            }
}
```

   Tip: To learn more about the syntax for the HTTP REST request, see the Remote.HTTP section of the ExtraHop Trigger API Reference.
3. Click **Save**.
4. In the HTTP table, hover over the **OK** status of the target to display connection activity.
   If the test is successful, the window displays the number of messages and bytes sent and received and the number of connection attempts, similar to the following figure:



Write the ODS trigger

In the following steps, you will write a trigger that specifies which metrics to send to the CloudWatch service and contains the command to send metric data through the open data stream.

Tip: As you build the trigger in this procedure, add comments that describe the purpose of a code snippet, restrictions, or best practices.

1. Click the ExtraHop logo in the upper left corner to return to the ExtraHop Web UI.
2. Click the System Settings icon, and then click **Triggers**.
3. Click **New** to open the Trigger Configuration window.
4. In the Name field, type `Metrics to CloudWatch`.
5. Click **Enable Debugging**.
6. In the Events field, select **HTTP_RESPONSE**.
7. Click the **Editor** tab.
8. Add the following trigger code to the editor to specify "ExtraHop" as a custom namespace for the metric data that will be displayed by the CloudWatch service:

```
let namespace = 'ExtraHop';
```

Note: The namespace value cannot be "AWS".

9. Add the following trigger code to the existing script to specify the name of the ODS target that you configured earlier:

```
let target = 'CloudWatch';
```

10. Add the following trigger code to the existing script to specify which metrics to will be transmitted to the CloudWatch service:

```
let metrics = [
    {
        'MetricName': 'processingTime',
```

```
        'Unit': 'Milliseconds',
        'Value' : HTTP.processingTime
    },
    {
        'MetricName': 'rspSize',
        'Unit': 'Bytes',
        'Value' : HTTP.rspSize
    }
];
```

11. Add the following trigger code to the existing script to specify the structure of the payload, which is defined by the `PutMetricData` method in the Amazon CloudWatch API:

```
let payload = 'Action=PutMetricData&Version=2010-08-01&Namespace=' +
 namespace;

let i,
    count = 0;

for (i = 0; i < metrics.length; i++) {
    let idx = i + 1,
        metric = metrics[i],
        val = metric.Value,
        attr;

    // If the metric value is NaN, do not publish.
    if (Number.isNaN(val)) {
        continue;
    }

    for (attr in metric) {
        payload += '&MetricData.member.' + idx + '.' +
            encodeURIComponent(attr) + '=' +
            encodeURIComponent(metric[attr]);
    }
    count++;
}

if (count == 0) {
    // No metrics to publish.
    return;
}
```

12. Add the following trigger code to the existing script to define the HTTP REST request that specifies the request path, headers, and payload:

```
let req = {
    'path': '/',
    'headers': {
        'Content-Type': 'application/x-www-form-urlencoded'
    },
    'payload': payload
};
```

This code is similar to the test request you performed in the previous procedure.

13. Add the following trigger code to the existing script to specify the ODS target and initiate the request to transmit metric data to that target:

```
Remote.HTTP(target).post(req);
```

14. Click **Save and Close**.

Assign the ODS trigger to a device

Before the trigger can send metric data to the CloudWatch service, you must assign the trigger to at least one device or network. For this walkthrough, you will assign the trigger to a single HTTP server in an activity group.

When creating your own triggers, assign triggers only to the specific devices that you need to collect metrics from to minimize the performance impact of your triggers on the ExtraHop system.

1. Click **Metrics** from the top menu.
2. From the left pane, click **Activity Groups**, and then click **HTTP Servers**.
3. From the HTTP Server table, select the checkbox for a single server that you know has web traffic.
4. From the Select Action drop-down menu, click **Assign Trigger**.
5. Click the checkbox next to the **Metrics to CloudWatch** trigger, and then click **OK**.

After the trigger is assigned, the system runs the trigger continuously until the trigger is disabled.

Verify data transmission to the ODS target

After the trigger has run, verify that data has been received by the ODS target, and then disable the trigger.

Important: Amazon Web Services is a tiered solution; there is no cost for the first tier unless usage is exceeded. Run this trigger for a short period of time to avoid exceeding the allowed amount of data. If you do not disable the trigger, and usage exceeds your allotted terms, you might incur additional costs.

1. Let the trigger run for 10-15 minutes.
2. Click the System Settings icon, and then click **Administration**.
3. From the System Configuration section, click **Open Data Streams**.
4. In the HTTP table, hover over the **OK** status of the CloudWatch target to display activity over the connection.

   If the trigger is successful, the window displays the number of messages and bytes sent and received, and the number of connection attempts.
5. Close the ExtraHop Admin UI window.
6. Click the System Settings icon, and then click **Triggers**.
7. In the **Triggers** table, click **Metrics to CloudWatch**.
8. Click **Disable trigger**.
9. Click **Save and Close**.

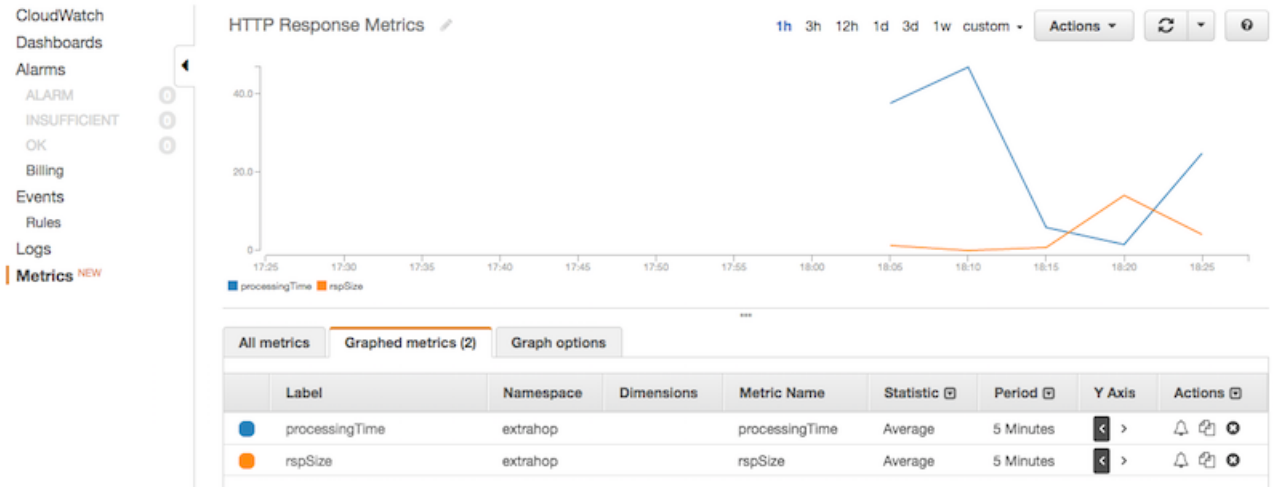
View results in AWS CloudWatch

After you have verified that metric data was sent to the ODS target, you can view the data with the CloudWatch service. In the following steps, you will find the metrics in CloudWatch and view the metric data on a graph.

1. Go to the Amazon Web Services site.
2. Click **Sign In to the Console** and enter your AWS login credentials.
3. From the list of AWS services, click **CloudWatch**.
4. From the left-hand menu, click **Metrics**.
   The All metrics tab displays the “ExtraHop” namespace created by the trigger and the “test” namespace created by the test request.
5. Click **ExtraHop**, and then click **Metrics with no dimensions**.

The tab displays the two metrics specified in the trigger, “processingTime” and “rspSize”.

6. Select the checkbox next to each metric to display metric data on the graph, similar to the following figure:



Next steps

Now that you have successfully sent metric data from your ExtraHop system to AWS CloudWatch, try modifying the trigger to send additional metrics or create a new ODS target to send data to other third-party tools.