Integrate IBM Node-Red with Splunk

Handbook

Version 0.1 (Draft)

**Revision History**

Note that Version 1.0 is the original draft of the reports design document. Each subsequent revision is assigned a version number, such as 1.1, 1.2 or, for major changes, 2.0.

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Version Date** | **Revision Summary** | **Author** |
| 0.1 | 10/06/2017 | Initial Draft | Nitin Kulkarni |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

[1 Introduction 4](#_Toc495059473)

[2 Configure Splunk 4](#_Toc495059474)

[3 Enabling Splunk from settings.js for local node-red instance and local Splunk running instance. 6](#_Toc495059475)

[4 Enabling Splunk from settings.js for node-red running under Docker/Container and/or local node-red instance and Splunk running on external web server. 8](#_Toc495059476)

[5 Enabling Splunk @Docker/Container Level (not using settings.js) 10](#_Toc495059477)

[6 Sample Test Result 10](#_Toc495059478)

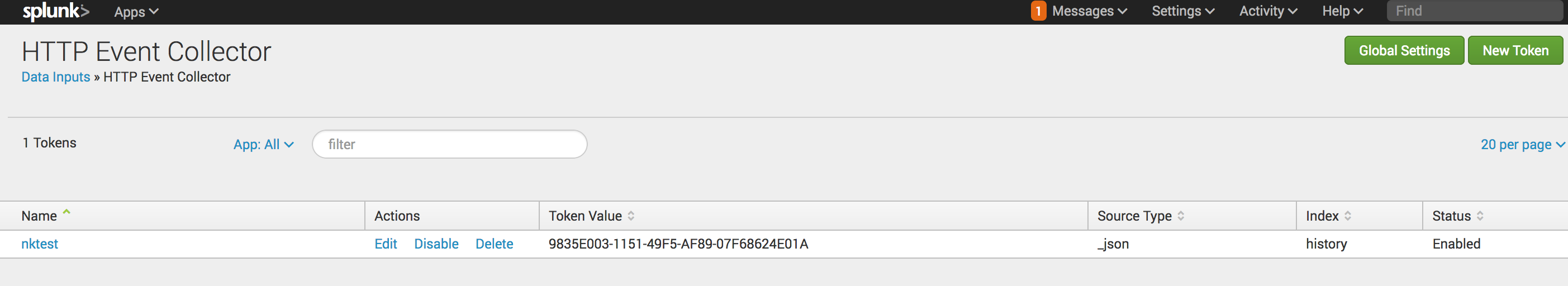
# **Introduction**

This document provides step by step instruction to integrate Splunk with Node-Red. Any exceptions thrown by Node-Red flow will be redirected to Splunk for the logging.

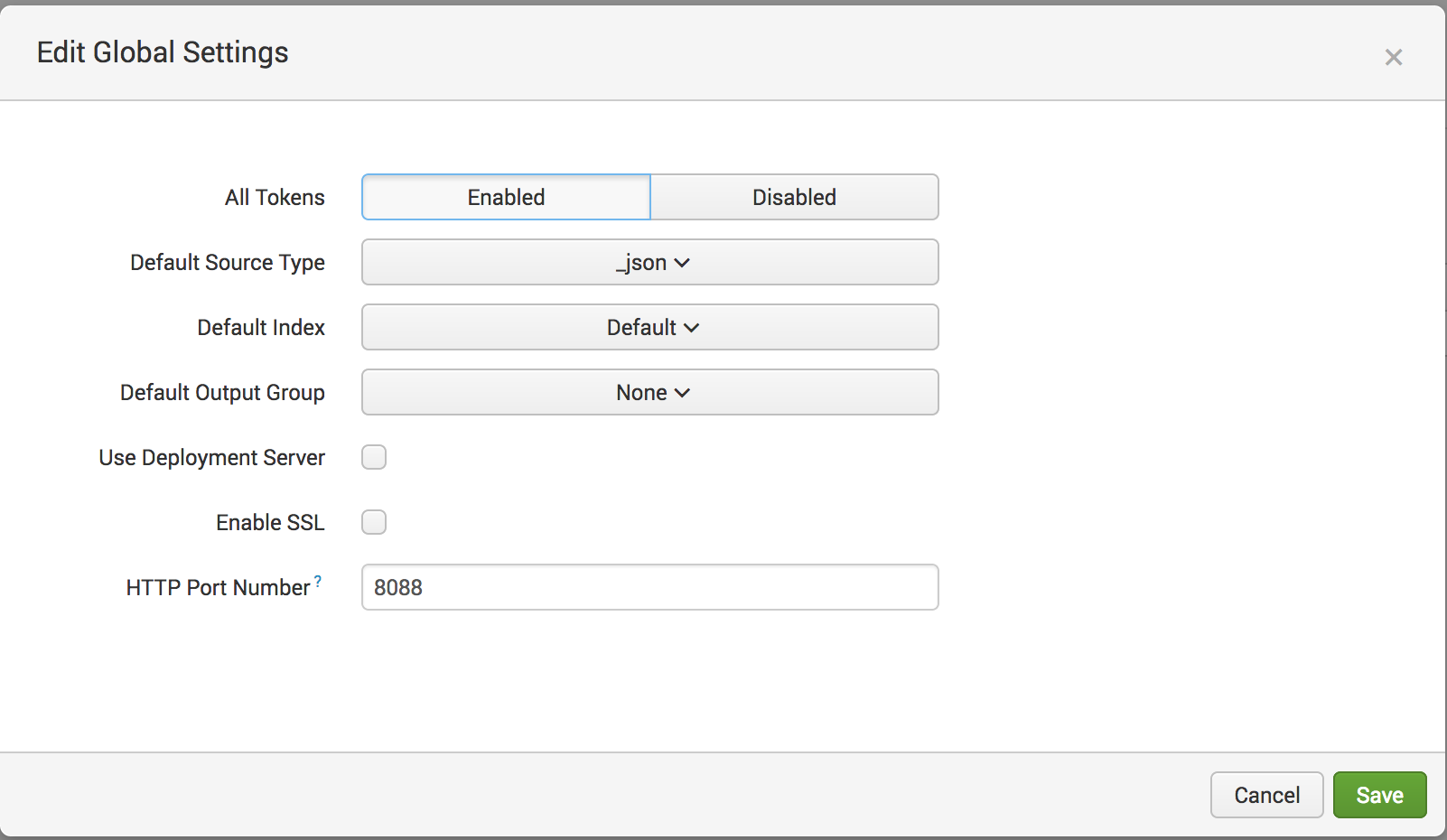
# **Configure Splunk**

This document assumes, Splunk is already configured and web server is running.

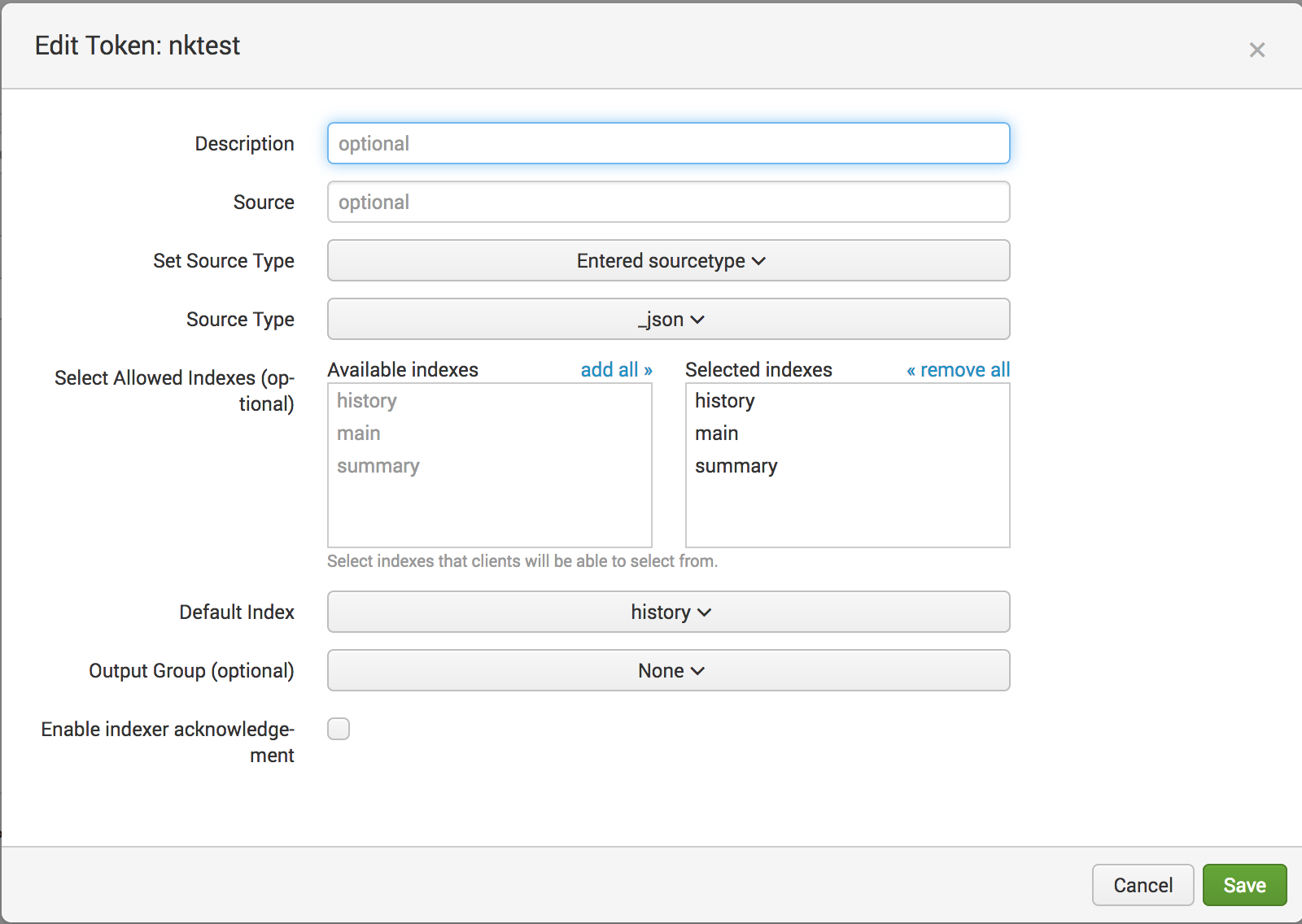
Ensure HTTP event collector is configured and active.



**Splunk Global Settings;**



**Splunk – HTTP event Collector Token Settings;**



Information required for enabling Splunk Logging in node-red:

|  |  |
| --- | --- |
| **Item** | **Sample Value** |
| HTTP Event collector URL | [http://localhost:8088/services/collector/eventTTP event collector](http://localhost:8088/services/collector/eventTTP%20event%20collector)  or  <http://72.139.178.89:8088/services/collector/event> |
| http Event Collector Token | "9835E003-1151-49F5-AF89-07F68624E01A"  or  "D8A01FCF-68FD-420B-BEBE-6666C19E9EF5" |

# **Enabling Splunk from settings.js for local node-red instance and local Splunk running instance.**

* Run ‘***npm install splunk-logging***’ from the shell prompt

Add following custom Splunk Logger in Settings.js

|  |
| --- |
| // Configure the logging output  logging: {  // Only console logging is currently supported  console: {  // Level of logging to be recorded. Options are:  // fatal - only those errors which make the application unusable should be recorded  // error - record errors which are deemed fatal for a particular request + fatal errors  // warn - record problems which are non fatal + errors + fatal errors  // info - record information about the general running of the application + warn + error + fatal errors  // debug - record information which is more verbose than info + info + warn + error + fatal errors  // trace - record very detailed logging + debug + info + warn + error + fatal errors  level: "debug",  // Whether or not to include metric events in the log output  metrics: false,  // Whether or not to include audit events in the log output  audit: false  }  ,  **// The following is custom Splunk Logger**  mySplunk: {  level:'debug',  metrics:true,  handler: function(conf) {  var splunk = require('splunk-logging');  var config = {  token: "9835E003-1151-49F5-AF89-07F68624E01A",  url: "http://localhost:8088/services/collector/event"  };  var splunk\_log = new splunk.Logger(config);  // Return the function that will do the actual logging  return function(msg) {  var payload = {  message : {  '@tags': ['UCG-node-red-Logging'],  '@timestamp': (new Date(msg.timestamp)).toISOString(),  '@msg-Level' : msg.level,  '@msg-content' : msg.msg  }  }  try {  splunk\_log.send(payload);  }catch(err) { console.log(err);}  }  }  } |

# **Enabling Splunk from settings.js for node-red running under Docker/Container and/or local node-red instance and Splunk running on external web server.**

* Run ‘***npm install splunk-logging***’ from the shell prompt in case of local node-red
* Run ‘***npm install splunk-logging***’ from the DockerFile if using Docker/Container

Add following custom Splunk Logger in Settings.js

|  |
| --- |
| // Configure the logging output  logging: {  // Only console logging is currently supported  console: {  // Level of logging to be recorded. Options are:  // fatal - only those errors which make the application unusable should be recorded  // error - record errors which are deemed fatal for a particular request + fatal errors  // warn - record problems which are non fatal + errors + fatal errors  // info - record information about the general running of the application + warn + error + fatal errors  // debug - record information which is more verbose than info + info + warn + error + fatal errors  // trace - record very detailed logging + debug + info + warn + error + fatal errors  level: "debug",  // Whether or not to include metric events in the log output  metrics: false,  // Whether or not to include audit events in the log output  audit: false  }  ,  **// The following is custom Splunk Logger**  mySplunk: {  level:'debug',  metrics:true,  handler: function(conf) {  var splunk = require('splunk-logging');  var config = {  token: "D8A01FCF-68FD-420B-BEBE-6666C19E9EF5",  url: "http:// 72.139.178.:8088/services/collector/event"  };  var splunk\_log = new splunk.Logger(config);  // Return the function that will do the actual logging  return function(msg) {  var payload = {  message : {  '@tags': ['UCG-node-red-Logging'],  '@timestamp': (new Date(msg.timestamp)).toISOString(),  '@msg-Level' : msg.level,  '@msg-content' : msg.msg  }  }  try {  splunk\_log.send(payload);  }catch(err) { console.log(err);}  }  }  } |

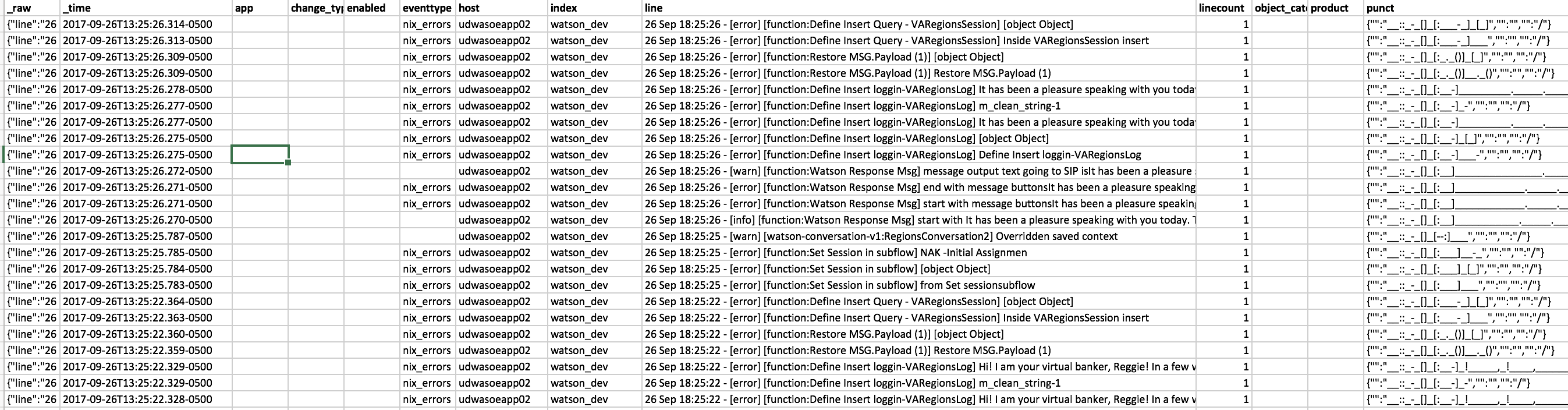
# **Enabling Splunk @Docker/Container Level (not using settings.js)**

In this case no changes are needed in setting.js

Include following parameters while creating Docker/Container.

|  |
| --- |
| docker run  -it  **--log-driver=splunk**  **--log-opt splunk-token=<splunk http event collector token>**  **--log-opt splunk-url=<splunk http event collector end-point>**  **--log-opt splunk-insecureskipverify=true.       <Use this option if SSL is enabled in Splunk Golbal Setting>**  **--log-opt tag="{{.Name}}/{{.FullID}}"**      -p 1880:1880   --name <container name>  <repository>/image name |

# **Sample Test Result**



Attached Splunk Extract.

