# DTCC Splunk Ingestion Work and Deliverable document



## Summary

ECD Will build a serverless Splunk Ingestion system for the Infrastructure Tools Management Team. We already built a client-side logging API that delivers log events to CloudWatch, and so we understand what it takes to push these same events to Splunk.

The purpose of this document is the list all roles and responsibilities across teams. It’s a living document and a statement of work that we can use to keep everyone on the same page.

## Deliverables

ECD will deliver the following artifacts

1. A Splunk Ingestion Lambda Function that will subscribe to one or more CloudWatch log groups and
2. A Configuration Lambda Function that will trigger whenever a new Log Group is created or deleted and automatically reconfigure the Splunk Ingestion Lambda function.
3. The necessary CloudWatch events (or AWS Config rule) used to trigger the configuration function
4. Terraform automation artifacts that can be handed to EHS for deployment.

Development will be done in Node.js for the ingestion code and Terraform for the automation.

## Features of the System

ECD Will Deliver on the following features. You can think of these as a mini set of business requirements:

1. The Splunk ingestion Lambda function will be configurable via environment variables. These variables drive the following:
   1. Splunk endpoint
   2. Splunk HEC token
   3. Splunk Index Name
2. The Splunk ingestion Lambda function should selectively prevent DEBUG and TRACE events from being sent to Splunk. This should be configurable as well.
3. The Splunk ingestion Lambda function should be able to send events to multiple indices. This means inspecting the payload and grouping events by the index it’s supposed to go to.
4. The Lambda Configuration Function will automatically configure any log group (add or delete) beginning with “**/DTCC/*SYSID***”

## Future Enhancements

Upon the completion of the base product we will discuss and consider the following additions

1. A Lambda function responsible for re-processing any failed events, names items pushed to the Lambda function dead letter queue.
2. Monitoring of the Splunk ingestion function for failed invocations

## Responsibilities

**ECD Will:**

1. Develop the initial software and ensure that it works correctly.
2. Develop the automation artifacts, specifically the Terraform ones.
3. Work with EHS to deploy the ingestion system
4. Provide level 2 support as necessary

**ITM Will:**

1. Own the software once it’s built
2. Provide immediate support for the system.
3. Provide future enhancements if ECD does not have the bandwidth.

**EHS Will:**

1. Build the Jenkins deployment pipelines using the automation artifacts produced by ECD.
2. Deploy the system to the required accounts
3. Engage with TRM to formally review the automation before going to PSE and Production