Version Description Document for R-ICMS: Regional Integrated Corridor Management System-Iteration 4

Version: 4.0

Approval date: 1/14/2021

DOCUMENT CONTROL PANEL		
File Name:	R-ICMS-VDD-3.1.docx	
File Location:	FDOT Sharepoint	
Version Number:	3.1	
	Name	**Date**
Created By:	Clay Weston	12/09/2020
Reviewed By:	Kevin Miller and John Horner	1/5/2021
	Clay Packard	1/9/2021
Modified By:		
Approved By:	Tushar Patel	1/14/2021

Table of Contents

List of Tables

List of Figures

List of Acronyms and Abbreviations

AAM	Active Arterial Management
AM	Ante Meridiem
API	Application Program Interface
AST	Agency for State Technology
ATMS	Advanced Traffic Management System
AVL	Automatic Vehicle Location
AWS	Amazon Web Services
CCTV	Closed Circuit Television
CDH	Cloudera Distributed Hadoop
CLI	Command Line Interface
OTS	Off the Shelf
CSV	Comma Separated Variable
DFE	Data Fusion Environment
DMS	Dynamic Message Signs
DOT	Department of Transportation
DSS	Decision Support System
ERD	Entity Relationship Diagram
ETL	Extract, Transform, Load
FCS	Florida Cybersecurity Standards
FDOT	Florida Department of Transportation
FTP/SFTP	File Transport Protocol / Secure File Transport Protocol
FQDN	Fully Qualified Domain Name
GIS	Geographic Information System
GTFS	General Transit Feed Specification
GTFS-RT	General Transit Feed Specification – Real Time
HCS7	Highway Capacity Software
HDFS	Hadoop Distributed File System
HTTPS	Hyper Text Transfer Protocol Secure
ICD	Interface Control Document
ID	Identifier
IEN	Information Exchange Network
IMC	Intersection Movement Counts
IT	Information Technology
ITS	Intelligent Transportation System
ITSIQA	Intelligent Transportation System Input Quality Assurance
JSON	JavaScript Object Notation
JWT	JSON Web Tokens
LDAP	Lightweight Directory Access Protocol
ME	Modeling Engine
MOE	Measure of Effectiveness
MS SQL	Microsoft SQL
MVC	Model View Controller
OAS	OpenAPI Specification
PD	Preliminary Design
PDF	Portable Document Format
PDR	Preliminary Design Review

PM	Post Meridiem
RCI	Roadway Characteristics Inventory
RDBMS	Relational Database Management System
REST	Representational State Transfer
R-ICMS	Regional Integrated Corridor Management System
RP	Response Plan
RPE	Response Plan Element
SDD	System Design Document
SHS	State Highway System
SLES	SUSE Linux Enterprise Server
SOT	Signal Optimization Tool
SQL	Structured Query Language
SSL	Secure Sockets Layer
TBD	To Be Determined
TGDC	Time Grouped Demand Cluster
TLS	Transport Layer Security
TSMO	Transportation Systems Management and Operations
UI	User Interface
UML	Unified Modeling Language
URL	Uniform Resource Locator
XML	Extensible Markup Language

1 Introduction

This document serves as the Version Description Document (VDD) for the Regional Integrated Corridor Management System (R-ICMS) software.

1.1 Overview

The R-ICMS is intended to be an initial implementation of a multi-modal regional transportation management system. The R-ICMS will integrate freeway, arterial, transit, and rail transportation management for the I-4 corridor, including management of transportation system components owned and operated by the State, as well as county, city, and regional transportation agencies.

The R-ICMS will consist of, but not be limited to, off-the-shelf (OTS) modeling software (provided by FDOT), a custom-built Decision Support System (DSS), a custom-built Information Exchange Network (IEN) subsystem that includes dashboards and other user interfaces to the system, and a Data Fusion Environment (DFE) to host data sources for both the R-ICMS and other external users and applications.

This project is funded and managed by District 5 of the Florida Department of Transportation (FDOT). It is intended for the use of District personnel, as well as personnel from the cities, counties, and transportation agencies located within the District. The initial deployment of the R-ICMS will be to the Transportation Management Center being built in District 5 by the FDOT.

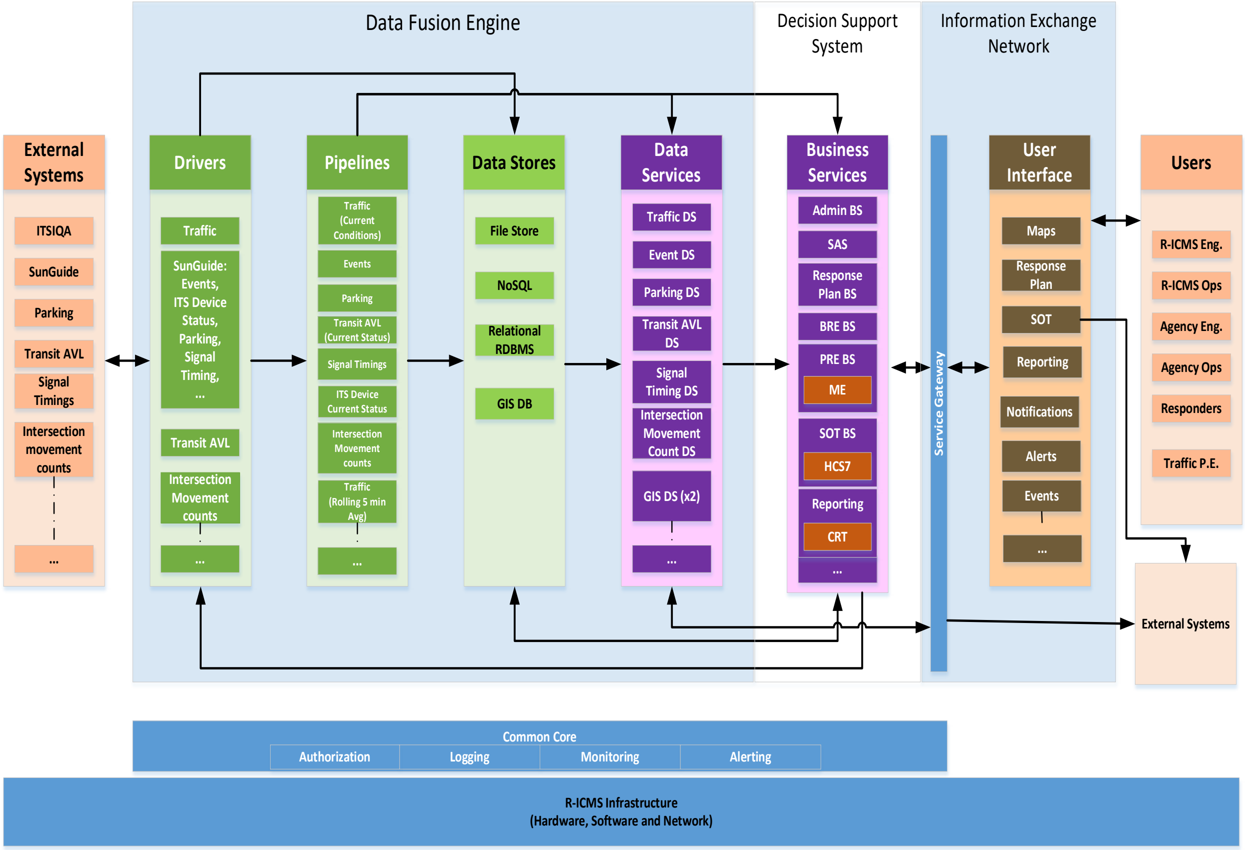


Figure 1 — High Level Architecture

2 Reference Documents

The following documents, of the exact issue shown, form a part of this document to the extent specified herein. In the event of a conflict between the documents referenced herein and the contents of this document, this document shall be considered the superseding requirement.

Table 1 — Reference Documents

Document Name	Document Location
R-ICMS-VDD-3.1-Placeholders.xlsx	FDOT R-ICMS Project SharePoint Site:
System and Subsystem Requirements Specification for R-ICMS for: Regional Integrated Corridor Management System: R-ICMS-REQ-1.0	FDOT R-ICMS Project SharePoint Site
BE521 - Executed Contract	Florida Department of Transportation D5prcustodian@dot.state.fl.us
BE521 – Contract Amendment 1	Florida Department of Transportation D5prcustodian@dot.state.fl.us
BE521 – Special Services 01 - 07	Florida Department of Transportation D5prcustodian@dot.state.fl.us
Data Sets Needed by ICMS - ICMS Requirements Table 7	FDOT R-ICMS Project SharePoint Site
Software Development Plan for the Regional Integrated Corridor Management System: R-ICMS-SDP-1.0	FDOT R-ICMS Project SharePoint Site
System Design Document for R-ICMS: Regional Integrated Corridor Management System: R-ICMS-SDD-4.0	FDOT R-ICMS Project SharePoint Site

3 Version Description

This version description document describes the installation and configuration of the R-CIMS software that was used during the R-ICMS Acceptance Test. The software was installed and configured on the production system at FDOT. Table 2 and Table 3 below show the configuration of the R-ICMS software in the production environment and related external resource dependencies. All hosts are on the D5-ITS.TSMO.DOT.STATE.FL.US domain. Site specific configuration values such as IP addresses and passwords are replaced with placeholders in the form of <PLACEHOLDER-NAME>. See the R-ICMS-VDD-3.1-Placeholders.xlsx document for actual values of these placeholders.

Table 2 - RICMS Production Physical Server Hosts

Role	Hostname	IP	OS	Rack	SPU Speed / Core	RAM	Disk	Notes
Cloudera edge 0	ITSSD5ICMSCDHE0	<HOST-CLOUDERA-EDGE-IP-0>	Ubuntu 18.04	1	Medium / High 2.1GHz, 12C/24T	256GB	2 x 1.2TB HDD 6 x 4TB HDD	
Cloudera master 0	ITSSD5ICMSCDHM0	<HOST-CLOUDERA-MASTER-IP-0>	Ubuntu 18.04	1	Medium / High 2.1GHz, 12C/24T	128GB	2 x 1.2TB HDD 8 x 1TB HDD	
Cloudera master 1	ITSSD5ICMSCDHM1	<HOST-CLOUDERA-MASTER-IP-1>	Ubuntu 18.04	2	Medium / High 2.1GHz, 12C/24T	128GB	2 x 1.2TB HDD 8 x 1TB HDD	
Cloudera worker 0	ITSSD5ICMSCDHW0	<HOST-CLOUDERA-WORKER-IP-0>	Ubuntu 18.04	1	Medium / High 2.1GHZ, 12C/24T	384GB	2 x 1TB HDD 10 x 4TB	
Cloudera worker 1	ITSSD5ICMSCDHW1	<HOST-CLOUDERA-WORKER-IP-1>	Ubuntu 18.04	2	Medium / High 2.1GHZ, 12C/24T	384GB	2 x 1TB HDD 10 x 4TB	
Cloudera worker 2	ITSSD5ICMSCDHW2	<HOST-CLOUDERA-WORKER-IP-2>	Ubuntu 18.04	3	Medium / High 2.1GHZ, 12C/24T	384GB	2 x 1TB HDD 10 x 4TB	
Cloudera worker 3	ITSSD5ICMSCDHW3	<HOST-CLOUDERA-WORKER-IP-3>	Ubuntu 18.04	1	Medium / High 2.1GHZ, 12C/24T	384GB	2 x 1TB HDD 10 x 4TB	
Cloudera Kafka 0	ITSSD5ICMSCDHK0	<HOST-CLOUDERA-KAFKA-IP-0>	Ubuntu 18.04	2	Medium / High 2.1GHz, 12C/24T	192GB	10 x 1TB HDD	

Cloudera Kafka 1	ITSSD5ICMSCDHK1	<HOST-CLOUDERA-KAFKA-IP-1>	Ubuntu 18.04	3	Medium / High 2.1GHz, 12C/24T	192GB	10 x 1TB HDD	
Cloudera utility 0	ITSSD5ICMSCDMU0	<HOST-CLOUDERA-UTILITY-IP-0>	Ubuntu 18.04	3	Medium / High 2.1GHz, 12C/24 T	128GB	2 x 1.2TB HDD 8 x 1TB HDD	
Elastic 0	ITSSD5ICMSES0	<HOST-ELASTICSEARCH-0>	Ubuntu 18.04	1	Medium / High 2.1GHZ, 12C/24T	64GB	2 x 1TB Raid 1 3 x 2.4TB	
Elastic 1	ITSSD5ICMSES1	<HOST-ELASTICSEARCH-1>	Ubuntu 18.04	2	Medium / High 2.1GHZ, 12C/24T	64GB	2 x 1TB Raid 1 3 x 2.4TB	
Elastic 2	ITSSD5ICMSES2	<HOST-ELASTICSEARCH-2>	Ubuntu 18.04	3	Medium / High 2.1GHZ, 12C/24T	64GB	2 x 1TB Raid 1 3 x 2.4TB	
Kubernetes Linux worker 0	ITSSD5ICMSKLW0	<HOST-KUBERNETES-WORKER-0>	Ubuntu 18.04	1	High /Medium 3GHz, 12C/24T	128GB	2 x 800 GB Raid 1	
Kubernetes Linux worker 1	ITSSD5ICMSKLW1	<HOST-KUBERNETES-WORKER-1>	Ubuntu 18.04	2	High /Medium 3GHz, 12C/24T	128GB	2 x 800 GB Raid 1	
Kubernetes Linux worker 2	ITSSD5ICMSKLW2	<HOST-KUBERNETES-WORKER-2>	Ubuntu 18.04	3	High /Medium 3GHz, 12C/24T	128GB	2 x 800 GB Raid 1	
Kubernetes Linux worker 3	ITSSD5ICMSKLW3	<HOST-KUBERNETE	Ubuntu 18.04	1	High /Medium 3GHz, 12C/24T	128GB	2 x 800 GB Raid 1	

		S-WORKER-3>						
Kubernetes Linux worker 4	ITSSD5ICMSKLW4	<HOST-KUBERNETES-WORKER-4>	Ubuntu 18.04	2	High /Medium 3GHz, 12C/24T	128GB	2 x 800 GB Raid 1	
Kubernetes Linux worker 5	ITSSD5ICMSKLW5	<HOST-KUBERNETES-WORKER-5>	Ubuntu 18.04	3	High /Medium 3GHz, 12C/24T	128GB	2 x 800 GB Raid 1	
Kubernetes master 0, Docker registry	ITSSD5ICMSKM0	<HOST-KUBERNETES-MASTER-0>	Ubuntu 18.04	1	Low / Few 3.8 GHZ, 2C/4T	16GB	2 x 1 TB Raid 1	Hosts Docker registry image-repo.ricms
Kubernetes master 1, Docker Swarm manager	ITSSD5ICMSKM1	<HOST-KUBERNETES-MASTER-1>	Ubuntu 18.04	2	Low / Few 3.8 GHZ, 2C/4T	16GB	2 x 1 TB Raid 1	Hosts OpenFaas
Kubernetes master 2	ITSSD5ICMSKM2	<HOST-KUBERNETES-MASTER-2>	Ubuntu 18.04	3	Low / Few e.g. G5500 3.8 GHZ, 2C/4T	16GB	2 x 1 TB Raid 1	
Kubernetes Windows worker 0, Swarm worker, Deploy host	ITSSD5ICMSKWW0	<HOST-SWARM-WORKER-0>	Windows Server 2019	1	High /Medium 3GHz, 12C/24T	128GB	2 x 800 GB Raid 1	Hosts OpenFaas functions
Kubernetes Windows worker 1, Swarm worker	ITSSD5ICMSKWW1	<HOST-SWARM-WORKER-1>	Windows Server 2019	2	High /Medium 3GHz, 12C/24T	128GB	2 x 800 GB Raid 1	Hosts OpenFaas functions

Kubernetes Windows worker 2, Authentication, Authorization, Windows API Caller	ITSSD5ICMSKWW2	<KUBERNETES-WINDOWS-WORKER-2>	Windows Server 2019	3	High /Medium 3GHz, 12C/24T	128GB	2 x 800 GB Raid 1	
Mongodb node 0	ITSSD5ICMSMDB0	<MONGO-NODE-0-IP>	Ubuntu 18.04	1	Medium / High 3.2GHz, 8C/16T	256GB	2 x 1TB Raid 1 10 x 10TBT	
Mongodb node 1	ITSSD5ICMSMDB1	<MONGO-NODE-1-IP>	Ubuntu 18.04	2	Medium / High 3.2GHz, 8C/16T	256GB	2 x 1TB Raid 1 10 x 10TBT	
Mongodb node 2	ITSSD5ICMSMDB2	<MONGO-NODE-2-IP>	Ubuntu 18.04	3	Medium / High 3.2GHz, 8C/16T	256GB	2 x 1TB Raid 1 10 x 10TBT	
Proxy for k8s	ITSSD5ICMSKP0	<KUBERNETES-PROXY>	Linux VM	N/A				
SQL Server	ITSSD5ICMSSQL1	<SDE-INSTANCE>	Windows VM	N/A				

Table 3 - RICMS Production External Resource Dependencies

Role	IP	Hostname	OS	Notes
D5 Aimsun (RPS)	<AIMSUN-RPS>	ITSSD5AIMSUN18	Linux	Aimsun hosted on port 8500
D5 Aimsun (SOT)	<AIMSUN-SOT>	ITSSD5AIMSUN03	Linux	Aimsun hosted on port 8500
SunStore ArcGIS Server	<ARCGIS-SUNSTORE>	ITSSD5ARCGIS02	Windows Server 2016	SunStore ESRI ArcGIS Server 10.7 https://arcserver.cflsmartroads.com:6443/arcgis/rest/services
ArcGIS Geoevent Server	<ARCGIS-GEOEVENT>	ITSSD5ARCGIS03	Windows Server 2016	ESRI ArcGIS Server Enterprise with GeoEvent Server geoevents.cflsmartroads.com (internal only for now)
GIS WebAdaptor	<GIS-WEBADAPTOR>	ITSSD5ARCGIS04	Windows	ESRI Web Adapter/Portal https://arcportal.cflsmartroads.com:7443/arcgis/home
GIS Desktop	<GIS-WORKSTATION>	ITSTD5ARCPRO01	Windows 10	ESRI ArcGIS Desktop & ArcPro Workstation
SMTP, for sending email only	N/A	TBD	N/A	SMTP running on port 25
SunGuide	<SUNGUIDE>	ITSSD5SG72DEV01	Windows	Advanced Traffic Management System hosted on port 8009, configured with TMDD C2C feed for traffic signals
TMDD C2C feed for traffic signals	<TMDD-C2C-TRAFFIC-SIGNALS>	ITSSD5TRFIS01	N/A	TMDD C2C feed for traffic signal data from a signal ATMS
SIIA	<SIIA>	ITSSD5MIMS-MA	N/A	Signalized Intersection Inventory Application API hosted on port 8092 https://mims.cflsmartroads.com/siia.api/api
ITSIQA	<ITSIQA>	ITSSD5SG72PR6	N/A	Traffic conditions Shared Folder for ITSIQA Files \\10.32.91.51\TSS-Share\ITSIQA\Web
Transit	<TRANSIT>	ITSSD5GTFSDA01	N/A	General Transit Feed System Aggregator API http://10.32.90.64/developer/api/v2
NWS	N/A	N/A	N/A	National Weather Service API https://api.weather.gov

ATSPM	<IP-ATSPM-DAT-DROP-SERVER>	ITSSD5RICMS01	N/A	ATSPM SMB drop folder for signal controller logs
Divas	N/A	https://DIVAS.cloud/VDS-API	N/A	CCTV hosted on DIVAS cloud

3.1 Inventory of Materials

This version of the R-ICMS includes the following:

- R-ICMS Source Code Version 4.0.0
- R-ICMS-REQ-4.0 Requirements Specification
- R-ICMS-PD-1.0 Preliminary Design Document
- R-ICMS-SDD-4.0 Iteration 4 System Design Document
- R-ICMS-STP-4.0 Iteration 4 Software Test Plan
- R-ICMS-TPD-4.0 Iteration 4 Test Procedures
- R-ICMS-STR-4.0 Iteration 4 Software Test Report

3.2 Inventory of Software Contents

The R-ICMS system is composed of custom microservices, Windows services, web applications, and executable applications developed exclusively for the R-ICMS system and Commercial-Off-The-Shelf (COTS) packages used in the development and installation of R-ICMS. The software contents of the initial iteration of R-ICMS are described in the following sections.

3.2.1 OTS Software Inventory

The OTS software used in R-ICMS, as well as in its development and installation is described in Table 2 — Inventory of OTS Software Packages.

Table 4 — Inventory of OTS Software Packages

Software	Version
Python	3.7.3
Kafka	2.4
.NET Core	2.1
ArcGIS Server	10.7
ArcGIS GeoEvent Server	10.7
ArcGIS Web Adaptor	10.7
HCS7	7.8.5
Kubernetes	1.12.3
Docker	17.03.3-ce (Kubernetes nodes, Ubuntu Linux) 18.06.1-ce (Swarm manager/registry, Ubuntu Linux) 18.09.9 (Windows)

Software	Version
Nginx	1.15.7
Linux for Kubernetes/Docker Swarm Nodes	Ubuntu Server 16.04.5 LTS
SQL Server	Enterprise 2016
Mongo DB	Enterprise 4.2.1
Windows	Windows Server 2016 Datacenter Windows Server 2019 Datacenter
Java	JDK 1.8.0_151_b12
Kibana	7.4.2+
Elasticsearch	7.4.2+
Filebeat	7.4.2+
Metricbeat	7.5.1+
faas-swarm	0.8.5+
faas-swarm-gateway	0.18.13+
faas-cli	0.11.0+
Hangfire	1.7.7+
MailKit	2.5.0+
PowerBI	1.8.7485.35104

3.2.2 Custom Software Inventory

The custom software developed for the R-ICMS system includes shared objects, Windows services, web applications, and executable applications. It is described in Table 3.

Table 5— Custom Software Components

Type	Name	Description
Driver	SunGuide Driver	Retrieves data from SunGuide databus and publishes the data to a Kafka topic
Driver	ITSIQA Current Driver	Retrieves data from an ITSIQA Current shared folder and publishes the data to a Kafka topic
Driver	ITSIQA Archive Driver	Retrieves data from an ITSIQA Archive shared folder and publishes the data to a Kafka topic
Driver	Signal Controller Log Driver	Retrieves data from the Signal Controller Log shared folder and publishes the data to a Kafka topic
Driver	SIIA Driver	Retrieves data from the SIIA RESTful API and publishes the data to a Kafka topic
Driver	GIS Static JSON Driver	Retrieves GIS Static data from the GIS RESTful API and publishes the data to a Kafka topic
Driver	GIS Static Shapefile Driver	Retrieves GIS Static Shapefile data from the Shapefile shared folder and publishes the data to a Kafka topic
Driver	Transit Real Time Archive Driver	Retrieves Transit Real Time Archive data from the GTFS Aggregator API and publishes the data to a Kafka topic
Driver	Transit Real Time Current Driver	Retrieves Transit Real Time Current data form the GTFS Aggregator API and publishes the data to a Kafka topic

Type	Name	Description
Driver	Transit Static Driver	Retrieves Transit Static data from the GTFS Aggregator API and publishes the data to a Kafka topic
Driver	Weather Alerts Driver	Retrieves weather alerts data from NWS API and publishes the data to a Kafka topic
Pipeline	GIS Static MongoDB Connector	Consumes GIS Static data from a Kafka topic and saves the data to MongoDB
Pipeline	SunGuide GeoEvent Connector	Consumes XML messages from a Kafka topic, transforms the XML, and publishes the transformed XML to a shared folder watched by a GeoEvent process
Pipeline	SunGuide JSON Processor	Consumes XML messages from a Kafka topic, converts the XML to JSON, and publishes the JSON to a Kafka topic
Pipeline	SunGuide MongoDB Connector	Consumes SunGuide JSON messages from a Kafka topic and saves the data to MongoDB
Pipeline	SunGuide Event MSSQL Connector	Consumes SunGuide Event JSON messages from a Kafka topic and synchronizes those events with MS SQL Server
Pipeline	ITSIQA GeoEvent Connector	Consumes XML messages from the ITSIQA all sources current traffic data topic, transforms the XML, and publishes the transformed XML data to a shared folder watched by a GeoEvent process
Pipeline	ITSIQA JSON Processor	Consumes XML messages from a Kafka topic, converts the XML to JSON, and publishes the JSON to a Kafka topic
Pipeline	ITSIQA MongoDB Connector	Consumes ITSIQA JSON messages from a Kafka topic and saves the data to MongoDB
Pipeline	ITSIQA Upstream Links MongoDB Connector	Consumes ITSIQA Current Link Config JSON data from a Kafka topic, creates a list of upstream links for each ITSIQA Link and saves the data to MongoDB
Pipeline	ITSIQA TMC Data for Aggregation MongoDB Connector	Consumes ITSIQA Archive TMC Data from a Kafka topic, reshapes the data to facilitate aggregation and saves the data to MongoDB
Pipeline	ITSIQA Archive MSSQL Connector	Consumes ITSIQA JSON messages from a Kafka topic, saves the timestamp data for each file to a MS SQL Server table, and stores first and last received timestamp for each ITSIQA Archive driver.
Pipeline	ITSIQA Archive Data Gap Processor	Consumes ITSIQA Archive data timestamps from MS SQL Server table, identifies the missing timestamp range and saves the missing data range to a MS SQL Server table
Pipeline	Signal Controller Log HDFS Connector	Consumes signal controller event codes as encoded DAT or decoded CSV messages from a Kafka topic, combines files with the same date into a single ZIP archive file, and writes the ZIP archive file to HDFS
Pipeline	Signal Controller Log DAT to CSV Processor	Consumes signal controller event codes as encoded DAT messages from a Kafka topic, decodes the DAT to CSV, and publishes the decoded CSV to a Kafka topic
Pipeline	Signal Controller Log CSV to JSON Processor	Consumes signal controller event codes as decoded CSV messages from a Kafka topic, converts the CSV to JSON, and publishes the JSON to a Kafka topic

Type	Name	Description
Pipeline	Signal Controller Log MongoDB Connector	Consumes signal controller event codes as JSON messages from a Kafka topic and saves the data to MongoDB; these are small files and high volume hence the separate pipeline.
Pipeline	SIIA MongoDB Connector	Consumes SIIA JSON messages from a Kafka topic and saves the data to MongoDB
Pipeline	SIIA JSON Processor	Consumes SIIA messages from a Kafka topic, transforms messages using SIIA Transformation rules and saves transformed JSON messages to a Kafka topic
Pipeline	HDFS Connector	Consumes multiple XML messages from a Kafka topic and writes them to a single file in HDFS
Pipeline	Transit Real Time GeoEvent Connector	Consumes Transit Real Time JSON messages from a Kafka topic and publishes as Geoevent XML files to a shared folder watched by a GeoEvent process
Pipeline	Transit Real Time MongoDB Connector	Consumes Transit Real Time messages from a Kafka topic and saves the data to MongoDB
Pipeline	Transit Static GeoEvent Connector	Consumes Transit Static data from a Kafka topic and publishes the data to a shared folder watched by a GeoEvent process
Pipeline	Transit Static GridFS Connector	Consumes Transit Static data from a Kafka topic and saves the data to MongoDB GridFS
Pipeline	Transit Static MongoDB Connector	Consumes Transit Static data from a Kafka topic and saves the data to MongoDB
Pipeline	Weather Alerts MongoDB Connector	Consumes Weather Alert data from a Kafka topic and saves the data to MongoDB
Data Service	Authorization Data Service	REST API providing access to roles, device groups, and the Active Directory user cache
Data Service	External Data Service	REST API providing access to ingested data
Data Service	Notification Data Service	REST API providing access to notifications
Data Service	Signal Optimization Tool Data Service	REST API providing access to optimization configurations and results
User Interface	User Interface	Provides a way for R-ICMS users to interact with the system
Business Service	Authentication Business Service	REST API providing JSON web token (JWT) generation and renewal
Business Service	Authorization Business Service	REST API for managing roles and device group mappings to Active Directory groups
Business Service	Data Quality Service	Provides REST APIs for recording gaps in data retrieval from a data source as well as archiving data schemas for data sources.
Business Service	Events Business Service	REST API to create and update Events from/to the database
Business Service	Monitoring Service	Provides REST APIs for creating and resolving alerts related to outages, unavailable data sources, invalid data from data sources, and fatal errors, .
Business Service	Notification Business Service	REST API to create, send, snooze, and resolve notifications and alerts
Business Service	Reports Service	Provides REST APIs for creating, updating, retrieving, and deleting report templates.

Type	Name	Description
Business Service	Response Plan Selection Service	Manages all business logic for evaluating and suggesting diversion routes around congestion related to SunGuide events. This service provides a REST API for data access and control flow for response plan selection, device approval, and response plan activation.
Business Service	Signal Optimization Tool Business Service	REST API to run signal optimizations
Business Service	SunGuide Broker Service	REST API to support adding comments, chronologies, and suggestion of response plans to SunGuide.
Business Service	TMC Clustering Service	Provides REST APIs to cluster intersection turn counts.
Business Service	Windows API Caller Business Service	REST API to call Active Directory APIs that must be run on a machine on the domain
Common Core	Active Directory Library	A library to wrap the low-level API calls that interact with Active Directory
Common Core	Authorization Library	A library that allows multiple services to easily check permissions and devices given a JWT
Common Core	Java Web Token Library	A library for interacting with JWTs that are stored in Redis and creating new JWTs.
Common Core	Logging Library	A library that sets up a common logging facility and format
Common Core	OpenAPI Library	A library that provides general purpose functionality to C# applications.
Common Core	Python Library	A library that provides general purpose functionality for python applications. This includes the definition of variable constants, logging functionality, and other base level functionality for Driver/Pipeline related applications.
Common Core	Redis Library	A library that simplifies interacting with the Redis in-memory database
Common Core	SunGuide Library	A library that supports the interaction with SunGuide. This includes web-socket functionality, conversion methods, and error definitions.
Common Core	User Cache Library	A library for creating and interacting with the cache of users and groups from Active Directory
Common Core	Utilities Library	A library that provides general purpose functionality to C# applications.
Common Core	Websockets Library	A library that supports streaming data to browsers using websockets

3.2.2.1 Deployment Folder Information

Host Name	IP	OS	Folder	Backup	Role / Purpose
ITSSD5ICMSKM0	<HOST-KUBERNETES-MASTER-0>	Linux	/etc/kubernetes/admin.conf	Yes	Kubernetes cluster config (for kubectl)
ITSSD5ICMSKWW0	<HOST-SWARM-WORKER-0>	Windows	C:\Deploy	Yes	Kubernetes configs for FDOT and deployment script Kibana monitoring watchers for FDOT are located in C:\Deploy\Monitoring\Watchers
ITSSD5ICMSKWW2	<KUBERNETES-WINDOWS-WORKER-2>	Windows	C:\Deploy\Docker\auth-docker-swarm	Yes	docker-compose.fdot.yml for authentication and authorization services
ITSSD5ICMSKWW2	<KUBERNETES-WINDOWS-WORKER-2>	Windows	C:\Deploy\WindowsApiCaller	Yes	Windows service for AD login (configured with FDOT API key)
ITSSD5ICMSKWW2	<KUBERNETES-WINDOWS-WORKER-2>	Windows	C:\Program Files\filebeat	Yes	filebeat executable and configuration
ITSSD5ICMSKP0	<KUBERNETES-PROXY>	Linux	/etc/haproxy/haproxy.cfg	Yes	Kubernetes master proxy configuration
ITSTD5ARCPRO01	<GIS-WORKSTATION>	Windows	C:\projects\ricms	Yes	GIS data folders

4 R-ICMS Installation Instructions

The following section provides an overview of how to install the R-ICMS software.

The R-ICMS software is installed using custom setup programs that are included in the Initial source code version and the manual operations described below.

4.1 Installation Overview

The following steps need to be performed to install the R-ICMS application software:

1. Prepare the R-ICMS Servers (see Section 4.3)
2. Install the database (see Section 4.4)
3. Install the application software (see Section 4.5)
4. Configure the system (see Section 4.6)

4.2 Platform Requirements

1. Kubernetes Cluster
    a. Operating System: Ubuntu Linux 18.04.4
    b. Docker 19.03.5
    c. Kubernetes 1.17.0
2. Docker Swarm Cluster
    a. Operating Systems
        i. Manager: Ubuntu Linux 18.04.4
        ii. Workers: Microsoft Windows Server 2019
    b. Docker 19.03.5
3. Cloudera Cluster
    a. Operating System: Ubuntu Linux 18.04.4
    b. Java Development Kit 1.8
    c. Cloudera Manager 6.3.3
    d. HDFS 3.0.0+cdh6.3.3
    e. Kafka 2.2.1+cdh6.3.3
4. MongoDB Server
    a. Operating System: Ubuntu Linux 18.04.4
    b. MongoDB Enterprise Server 4.2.1
5. Microsoft SQL Server
    a. Operating System: Microsoft Windows Server 2016
    b. SQL Server 2016
6. GIS Environment
    a. Portal for ArcGIS
        i. Operating System: Microsoft Windows Server <version>
        ii. Portal for ArcGIS 10.7.1, Web Adaptor for Portal 10.7.1
    b. ArcGIS Server
        i. Operating System: Microsoft Windows Server <version>
        ii. ArcGIS Server 10.7.1, Web Adaptor for Server 10.7.1
    c. GeoEvent Server
        i. Operating System: Microsoft Windows Server 2016 Datacenter
        ii. ArcGIS Server 10.7.1, GeoEvent Server 10.7.1, Web Adaptor for Server 10.7.1
    d. GIS workstation with ArcGIS Desktop
        i. Operating System: Microsoft Windows 10
        ii. ArcGIS Desktop 10.7.1

7. Kubernetes Control Plane Proxy
    a. Operating system: Ubuntu Linux 18.04.4
    b. HA-Proxy 1.8.8-1ubuntu0.9
8. Elasticsearch Cluster
    a. Operating System: Ubuntu 18.04.3
    b. Elasticsearch 7.4.2
9. Authorization Server
    a. Operating System: Microsoft Windows Server 2019
    b. Docker 19.03.5

4.3 Server Preparation

4.3.1 Cloudera Nodes Preparation

The Cloudera Stack will be installed on the following hosts:

1. <HOST-CLOUDERA-UTILITY-0>
2. <HOST-CLOUDERA-EDGE-0>
3. <HOST-CLOUDERA-MASTER-0>
4. <HOST-CLOUDERA-MASTER-1>
5. <HOST-CLOUDERA-KAFKA-0>
6. <HOST-CLOUDERA-KAFKA-1>
7. <HOST-CLOUDERA-WORKER-0>
8. <HOST-CLOUDERA-WORKER-1>
9. <HOST-CLOUDERA-WORKER-2>
10. <HOST-CLOUDERA-WORKER-3>

4.3.1.1 Hardware Configuration

First, the physical servers need to be configured to setup disks for OS installation and data storage.

Hardware configuration using BIOS. Use the hardware manufacturer instructions to configure the hard disks with the following configuration for each of the nodes as described below:

1. Configure hard disks
    a. For all nodes: Setup a 2TB RAID 0 volume for OS installation.
    b. For Cloudera Manager Service on edge node, setup 6 x 4TB RAID 10 volumes for data storage.
    c. For Name node and HDFS Checkpoint Directories on master nodes, setup 2 x 1TB RAID 1 volumes.

d. For Kafka data directories on Kafka hosts, setup 8 x 1TB RAID 10 volumes for data directories.

e. For Zookeeper data directory on utility node and master nodes, setup 1 x 1TB RAID 0 volumes.

f. For Journal node edits directory on master nodes, setup 1 x 1TB RAID 0 volumes.

g. For MariaDB data directory and Journal node edits directory on utility node, setup 6 x 1TB RAID 10 volumes.

h. For all Worker nodes, setup individual disks (NO RAID).

4.3.1.2 OS Installation and Configuration

4.3.1.2.1 OS Installation

After the physical hardware is configured, the next step is to install the Operating System on the nodes. OS installation will be the same for all nodes in the cluster.

1. Download the Ubuntu Server 18.04.4 LTS image and prepare a bootable flash drive.

   http://old-releases.ubuntu.com/releases/18.04.4/ubuntu-18.04-server-amd64.iso

   Use the Rufus tool (https://rufus.ie/) to create a bootable flash drive from the Ubuntu Server 18.04.4 LTS image that was downloaded.

2. Plugin the flash drive into a USB port and boot the machine from the flash drive, which will guide through the installation process. Note: Booting from Flash Drive may require a change to the boot sequence in the BIOS settings. This will launch the Ubuntu Server 18.04.4 LTS Installer.

3. Set up the OS disk as logical volume group and proceed with the installation. Allocate at least 300 MB for file system.

4. When prompted, create a user named administrator. Ubuntu will automatically grant this user sudo privileges (i.e. to run commands as a superuser)

4.3.1.2.2 OS Configuration

For all nodes, perform the following actions to partition, format, and mount the data drives/volumes (and only the data drives) on each node.

Delete any existing partitions

```
$ sudo fdisk /dev/sd<letter>
```

In the fdisk menu choose d (delete) then w (write)

Create a single partition for the entire disk for formatting as ext4

```
$ sudo parted -a opt /dev/sd<letter> mkpart primary ext4 0% 100%
```

Make the ext4 file system on the partition

```
$ sudo mkfs.ext4 -L datapartition /dev/sda<letter> < /usr/bin/yes
```

Create the mount point

```
$ sudo mkdir /data<number>
```

Assign <number> according to rank of <letter>. For example, if <letter> = a then assign <number> = 1.

As sudoer edit ***/etc/fstab*** and add the following line

```
/dev/sd<letter> /data<number> ext4 defaults,noatime 0
```

Mount the new volume

```
$ sudo mount –a
```

Check if volume is mounted

```
$ df
```

In the df output look for line where Filesystem value is ***/dev/sd<letter>*** (what was just added to ***/etc/fstab***) and the corresponding Mounted On value is correct.

Repeat the steps above for each data drive.

Setup the administrator user with password-less access (ssh) among all the nodes. The administrator user from one node must be able to login to all other nodes without using password with ssh in order to install Cloudera software. Follow the steps below to set up password-less access among the nodes:
ssh to a node as administrator user

Create the RSA Key Pair

```
$ ssh-keygen -t rsa
```

Accept defaults for the prompts EXCEPT for the passphrase if a passphrase is desired

The public key will be stored in ***/home/administrator/.ssh/id_rsa.pub***

Copy the Public Key to the other nodes of the cluster

```
ssh-copy-id administrator@<remote host>
```

Test the ssh connectivity to the remote host

```
ssh <user>@<remote host>
```

Repeat the above steps on each node of the cluster

4.3.1.2.2.1 Network Interface Bonding

1. Log directly into the node to be configured as the administrator user.
2. Run ***ifconfig*** and make note of the interfaces with a value of Link ***encap:Ethernet***. In subsequent steps eno1 and eno2 will be used.
3. Turn off networking: ***$ sudo stop networking***

4. Back up `/etc/network/interfaces`:

   ```
   $ sudo cp /etc/network/interfaces /etc/network/interfaces.bak
   ```

5. Edit `/etc/network/interfaces` and replace the content with the following, replacing the placeholders with the appropriate values for the node being configured.

   ```
   # This file describes the network interfaces available on your system
   # and how to activate them. For more information, see interfaces(5).

   source /etc/network/interfaces.d/*

   # The loopback network interface
   auto lo
   iface lo inet loopback

   # The primary network interface
   auto eno1
   iface eno1 inet manual
   bond-master bond0

   auto eno2
   iface eno2 inet manual
   bond-master bond0

   auto bond0
   iface bond0 inet static
   address <host IP>
   gateway <gateway IP>
   netmask <host netmask>
   dns-nameservers <dns IP>
   bond-mode 0
   bond-slaves eno1 eno2
   ```

6. Turn on networking: `$ sudo start networking`
7. Run `ifconfig` and verify that there is now an interface named bond0 that is configured with the correct host IP.
8. Repeat the above steps on each node of the cluster.

4.3.1.2.2.2 Update and Upgrade System package index

Before proceeding with installation and other configuration, update and upgrade all system packages to have smooth installation process

```
$ sudo apt-get update && sudo apt-get upgrade -y
```

4.3.1.2.2.3 Configure Network Names

Configure each host in the cluster as follows to ensure that all members can communicate with each other:

<u>Login and run commands as the administrator user</u>

1. Set the host name using the below command

```
$ sudo hostnamectl set-hostname $(hostname).d5-its.tsmo.dot.state.fl.us
```

2. Verify the hostname using the below command

```
$ uname –a
```

3. Edit `/etc/hosts` with the IP address and fully qualified domain name (FQDN) of each host in the cluster. You can add the unqualified name as well.

```
$ sudo cp /etc/hosts /etc/hosts.bak
cat <<EOF | sudo tee /etc/hosts > /dev/null
127.0.0.1       localhost
<HOST-CLOUDERA-EDGE-IP-0>   <HOST-CLOUDERA-EDGE-FQDN-0>   <HOST-CLOUDERA-EDGE-0>
<HOST-CLOUDERA-KAFKA-IP-0>   <HOST-CLOUDERA-KAFKA-FQDN-0>   <HOST-CLOUDERA-KAFKA-0>
<HOST-CLOUDERA-KAFKA-IP-1>   <HOST-CLOUDERA-KAFKA-FQDN-1>   <HOST-CLOUDERA-KAFKA-1>
<HOST-CLOUDERA-MASTER-IP-0>  <HOST-CLOUDERA-MASTER-FQDN-0>  <HOST-CLOUDERA-MASTER-0>
<HOST-CLOUDERA-MASTER-IP-1>  <HOST-CLOUDERA-MASTER-FQDN-1>  <HOST-CLOUDERA-MASTER-1>
<HOST-CLOUDERA-UTILITY-IP-0> <HOST-CLOUDERA-UTILITY-FQDN-0> <HOST-CLOUDERA-UTILITY-0>
<HOST-CLOUDERA-WORKER-IP-0>  <HOST-CLOUDERA-WORKER-FQDN-0>  <HOST-CLOUDERA-WORKER-0>
<HOST-CLOUDERA-WORKER-IP-1>  <HOST-CLOUDERA-WORKER-FQDN-1>  <HOST-CLOUDERA-WORKER-1>
<HOST-CLOUDERA-WORKER-IP-2>  <HOST-CLOUDERA-WORKER-FQDN-2>  <HOST-CLOUDERA-WORKER-2>
<HOST-CLOUDERA-WORKER-IP-3>  <HOST-CLOUDERA-WORKER-FQDN-3>  <HOST-CLOUDERA-WORKER-3>
```

```
# The following lines are desirable for IPv6 capable hosts
::1     localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
EOF
```

4. Verify the host file.

```
$ cat /etc/hosts
```

4.3.1.2.2.4 Disabling the Firewall

To disable the firewall on each host in your cluster, perform the following step on each node of the cluster as administrator user.

```
$ sudo service ufw stop
```

4.3.1.2.2.5 Enable an NTP Service

CDH requires that you configure a Network Time Protocol (NTP) service on each machine in your cluster. Perform the following steps on each node of the cluster as administrator user.

1. Install the ntp package

```
$ sudo apt-get install -y ntp ntpdate
```

2. Edit the `/etc/ntp.conf` file to add NTP servers

```
$ cat <<EOF | sudo tee -a /etc/ntp.conf > /dev/null
server 0.pool.ntp.org
server 1.pool.ntp.org
server 2.pool.ntp.org
EOF
```

3. Start the ntp service

```
$ sudo service ntp start
```

4. Configure the ntp service to run at boot

```
$ sudo systemctl enable ntp
```

5. Check the ntp service status and ensure it is active and enabled.

```
$ sudo service ntp status
```

6. Disable built-in synchronization service from timedatectl

```
$ sudo timedatectl set-ntp no
```

7. Synchronize the system clock to the NTP server

```
$ sudo ntpdate -u 0.pool.ntp.org
```

8. Synchronize the hardware clock to the system clock

```
$ sudo hwclock --systohc
```

4.3.1.2.2.6 Install Python 2.7 and pscyopg2

While Cloudera only requires Python 2.7.5 or lower to be installed on hosts running Hue, Python is generally useful so it will be installed on all nodes.

1. Install Python 2.7

```
$ sudo apt install -y python2.7 python-pip
```

2. Install Python PostgreSQL Database Adapter

```
$ sudo pip install psycopg2==2.7.5 --ignore-installed
```

3. Verify that Python 2.7 is installed

```
$ python –version
```

4.3.1.2.2.7 Set vm.swappiness Linux Kernel Parameter

The Linux kernel parameter, vm.swappiness, is a value from 0-100 that controls the swapping of application data (as anonymous pages) from physical memory to virtual memory on disk. The higher the value, the more aggressively inactive processes are swapped out from physical memory. The lower the value, the less they are swapped, forcing filesystem buffers to be emptied.

On most systems, vm.swappiness is set to 60 by default. This is not suitable for Hadoop clusters because processes are sometimes swapped even when enough memory is available. This can cause lengthy garbage collection pauses for important system daemons, affecting stability and performance.

Cloudera recommends that you set vm.swappiness to a value between 1 and 10, preferably 1, for minimum swapping on systems. Run below command on each host.

```
$ sudo sysctl -w vm.swappiness=1
$ echo vm.swappiness=1 | sudo tee -a /etc/sysctl.conf >
/dev/null
```

4.3.1.2.2.8 Install Java Development Kit

Run below command to install OpenJDK

```
$ sudo apt-get install -y openjdk-11-jdk
```

Verify the Java Version installed

```
$ java -version
```

4.3.1.2.2.9 Install the MySQL JDBC Driver

MySQL JDBC driver is required for a role to access the Database. Run the below command on each host to install MySQL JDBC driver.

```
$ sudo apt-get install -y libmysql-java
```

4.3.1.2.2.10 Create a root account

To configure Auto-TLS on Cloudera, we need to create root account as a workaround solution for login issues. Run the below commands on each node of the Cloudera cluster as administrator or a human user with sudo privileges.

1. SSH to the node
2. Change Password for root user

   ```
   $ sudo passwd root
   ```

   #Type <CLOUDERA-ROOT-PASSWORD> twice
3. Update sshd config to allow root user to ssh

   ```
   $ sudo sed -i 's/#PermitRootLogin prohibit-password/PermitRootLogin yes/' /etc/ssh/sshd_config
   ```
4. Restart ssh service

   ```
   $ sudo service ssh restart
   ```
5. Test if you are able to ssh using root user

4.3.2 Elasticsearch Nodes Preparation

The Elasticsearch server will be installed on the following hosts to create a 3-node cluster:

11. <HOST-ELASTICSEARCH-0>
12. <HOST-ELASTICSEARCH-1>
13. <HOST-ELASTICSEARCH-2>

4.3.2.1 Hardware Configuration

Before installing the operating system, configure the disks on each of the hosts as follows in the following order:

2x1TB Raid1 Virtual Disk – Operating System Disk

3x2.4TB RAID0 Virtual Disk – Data Disk

4.3.2.2 OS Installation and Configuration

Perform the following actions to install the operating system and to partition, format, and mount the data drives/volumes on each Elasticsearch node.

Install the Ubuntu 18.04.4 operating system on the operating system disk.

The following steps depend on the number of non-OS reserved `/dev/sd<letter>` drives where `<letter>` denotes a single character that defines a different drive. For each available `sd` drive, create an ext4 filesystem.

```
$ sudo mkfs.ext4 "/dev/sd<letter>"
```

Create a new directory called `data` under the root directory.

```
$ sudo mkdir /data
```

For each ext4 filesystem created, create a sub directory under `/data` to be used as a mount point. The name of the sub directory is a two-digit number (`<number>`) that corresponds to the rank of the letter in `sd<letter>`. For example, if the first available `sd` drive is `/dev/sdb`, then the corresponding mount point would be `/data/00`.

```
$ sudo mkdir /data/<number>
```

For each ext4 filesystem and corresponding mount point directory, add an entry to `/etc/fstab`.

```
/dev/sd<letter>   /data/<number>    ext4  defaults    0     0
```

Mount the new file systems

```
$ sudo mount -a
```

4.3.3 MongoDB Server

MongoDB server will be installed on the following hosts to create a 3-node replica set:

<HOST-MONGODB-0>

<HOST-MONGODB-1>

<HOST-MONGODB-2>

4.3.3.1 Hardware Configuration

Before installing the operating system, configure the disks on each of the hosts as follows in the following order:

1. 2x1TB RAID1 Virtual Disk – Operating System Disk
2. 10x10TB RAID10 Virtual Disk – Data Disk

4.3.3.2 OS Installation and Configuration

1. Install the Ubuntu 18.04.4 operating system on the operating system disk.
2. Create a filesystem on the data disk and mount the data disk.
    a. Remove any existing partition tables
    ```
    sudo wipefs --all --force /dev/sdb
    ```
    b. Create an ext4 filesystem
    ```
    sudo mkfs.ext4 /dev/sdb
    ```
    c. Create a mount point

```
sudo mkdir -p /data/mongo
```

   d. Add the following line to /etc/fstab

```
/dev/sdb /data/mongo ext4 defaults 0 0
```

   e. Mount /data/mongo

```
mount –a
```

3. Configure swappiness
   a. Set the running value

```
sudo sysctl vm.swappiness=1
```

   b. Set the configured value
      i. Open /etc/sysctl.conf for editing
      ii. Ensure there is a line as follows and save the file

```
vm.swappiness = 1
```

4.3.4 Kubernetes Servers

The Kubernetes server will be installed on the following hosts:

1. <HOST-KUBERNETES-MASTER-0>
2. <HOST-KUBERNETES-MASTER-1>
3. <HOST-KUBERNETES-MASTER-2>
4. <HOST-KUBERNETES-WORKER-0>
5. <HOST-KUBERNETES-WORKER-1>
6. <HOST-KUBERNETES-WORKER-2>
7. <HOST-KUBERNETES-WORKER-3>
8. <HOST-KUBERNETES-WORKER-4>
9. <HOST-KUBERNETES-WORKER-5>

4.3.4.1 Hardware Configuration

Before installing the operating system, configure the disks on each of the master nodes as follows (Note: there are no data disks for the Kubernetes nodes):

1. 2x1TB NO RAID Virtual Disk – Operating System Disk (Only 1 TB is used for OS)

Configure the disks on each of the worker nodes as follows:

1. 2x960GB RAID1 Virtual Disk – Operating System Disk

4.3.4.2 OS Installation and Configuration

Perform the following actions to install the operating system and to prepare for the installation of Kubernetes on each Kubernetes node.

1. Add the following entry to the hosts file located at /etc/hosts:
   ```
   <IMAGE-REPO-IP> image-repo.ricms
   ```
2. Set `bridge-nf-call-iptables` to 1
   ```
   sudo sysctl net.bridge.bridge-nf-call-iptables=1
   ```
3. Increase the amount of inotify watchers
   a. Set the `max_user_watches` to a higher value
      ```
      sudo echo "fs.inotify.max_user_watches=524288" | sudo tee -a
      /etc/sysctl.conf
      ```
   b. Load the settings from `/etc/sysctl.conf`
      ```
      sudo sysctl –p
      ```
   c. Verify that the value was updated to 524288
      ```
      cat /proc/sys/fs/inotify/max_user_watches
      ```
4. Enable devices for swapping
   ```
   sudo swapoff -a
   sed -re '/\sswap\s/s/^([^#])/#\1/' -i /etc/fstab
   ```
5. Reboot the machine

4.3.5 Docker Swarm Servers

The Swarm cluster will be installed on the following hosts:

1. <HOST-KUBERNETES-MASTER-1>
2. <HOST-SWARM-WORKER-0>
3. <HOST-SWARM-WORKER-1>

4.3.5.1 Linux Manager Node

Docker should be installed on the <HOST-SWARM-MANAGER-0> following the instructions per Section 4.3.3; if the host is the same as <HOST-KUBERNETES-MASTER-1> you do not need to repeat that process. The commands in this section are to be run on the Swarm manager in a Linux shell as a user with sudo privileges.

Initialize the Swarm cluster:

```
$ sudo docker swarm leave –force
$ sudo docker swarm init --advertise-addr=<HOST-KUBERNETES-MASTER-1> --
listen-addr <HOST-SWARM-MANAGER-1>:2377
$ sudo docker swarm join-token worker
```

Record the join token `<SWMTKN>` from the last command for use on the worker nodes. Note, some worker nodes must be labeled by the managed node after they are joined to the cluster, which is described in the worker node configuration sections.

Create a directory for installation of OpenFaas:

```
$ cd /opt
$ sudo mkdir open-faas
```

```
$ sudo chgrp <GROUP-RICMS-ADMIN> open-faas/
$ cd open-faas/
$ sudo chmod g+w .
$ sudo chmod g+s .
$ sudo setfacl -d -m g::rw .
$ getfacl .
```

Verify the output of the getfacl command:

```
# file: .
# owner: root
# group: <GROUP-RICMS-ADMIN>
# flags: -s-
user::rwx
group::rwx
other::r-x
default:user::rwx
default:group::rw-
default:other::r-x
```

Install OpenFaaS and its command line utility, faas-cli:

```
$ curl -sSL https://cli.openfaas.com | sudo -E sh
$ git clone https://github.com/openfaas/faas
$ cd faas
```

Edit docker-compose.yml, add/update the following fields:

- queue-worker. environment.ack_wait: `"<OPENFAAS-MAX-WAIT>"`
- queue-worker.deploy.replicas: `<OPENFAAS-MAX-REPLICA>`
- gateway.environment.readTimeout: `"<OPENFAAS-MAX-WAIT>"`
- gateway.environment.writeTimeout: `"<OPENFAAS-MAX-WAIT>"`
- gateway.environment.upstreamTimeout: `"<OPENFAAS-MAX-UPSTREAM>"`

Deploy the OpenFaas services in Docker:

```
$ sudo ./deploy-stack.sh
```

The deploy_stack.sh command will output a command to log in to the deployment. Copy and paste this command in the same terminal and run it, e.g.

```
$ echo -n <password> | faas-cli login --username=admin --password-stdin
```

Create a directory for OpenFaas user defined functions:

```
$ cd /opt
$ sudo mkdir functions
$ sudo chgrp <GROUP-RICMS-ADMIN> functions/
$ cd functions/
$ sudo chmod g+w .
$ sudo chmod g+s .
$ sudo setfacl -d -m g::rw .
$ getfacl .
```

Verify the output of the getfacl command:

```
# file: .
# owner: root
# group: <GROUP-RICMS-ADMIN>
# flags: -s-
```

```
user::rwx
group::rwx
other::r-x
default:user::rwx
default:group::rw-
default:other::r-x
```

4.3.5.2 Windows Worker Nodes

All `PS>` commands listed in this section **should run in PowerShell as Administrator**. These steps should be done on all Swarm worker nodes.

Verify the operating system build is later than 14393.1083, required for Docker in Swarm mode:

```
PS> systeminfo.exe | Select-String "Version"
```

You must upgrade the system to the required minimum version before proceeding with the next steps.

Add the following line to C:\Windows\system32\drivers\etc\hosts file:

```
<IMAGE-REPO-IP> image-repo.ricms
```

The firewall must be disabled, or the following ports must be open for Docker Swarm:

- TCP port 2377 for cluster management communications
- TCP and UDP port 7946 for communication among nodes
- UDP port 4789 for overlay network traffic

Verify firewall is disabled or use the commands below to open ports required for Docker Swarm:

```
# open incoming ports
PS> netsh advfirewall firewall add rule name="docker_cluster" dir=in
action=allow protocol=TCP localport=2377
PS> netsh advfirewall firewall add rule name="docker_swarm_tcp" dir=in
action=allow protocol=TCP localport=7946
PS> netsh advfirewall firewall add rule name="docker_swarm_udp" dir=in
action=allow protocol=UDP localport=7946
PS> netsh advfirewall firewall add rule name="docker_overlay" dir=in
action=allow protocol=UDP localport=4789

# open outgoing ports
PS> netsh advfirewall firewall add rule name="docker_cluster" dir=out
action=allow protocol=TCP localport=2377
PS> netsh advfirewall firewall add rule name="docker_swarm_tcp" dir=out
action=allow protocol=TCP localport=7946
PS> netsh advfirewall firewall add rule name="docker_swarm_udp" dir=out
action=allow protocol=UDP localport=7946
PS> netsh advfirewall firewall add rule name="docker_overlay" dir=out
action=allow protocol=UDP localport=4789

# verify above
PS> Get-NetFirewallRule | where {$_.DisplayName -like 'docker*'}
```

4.3.5.2.1 Install and Configure Docker EE and Compose

Enable Windows to run containers and install Docker EE:

```
PS> Install-WindowsFeature containers
PS> Restart-Computer
PS> Install-Module DockerProvider -Force
      Install NuGet provider: Y
PS> Install-Package Docker -ProviderName DockerProvider -Force -
RequiredVersion 18.09

# test if Docker is running
PS> docker images
```

Install docker-compose, which requires TLS 1.2:

```
PS> [Net.ServicePointManager]::SecurityProtocol =
[Net.SecurityProtocolType]::Tls12
PS> Invoke-WebRequest
"https://github.com/docker/compose/releases/download/1.24.0/docker-
compose-Windows-x86_64.exe" -UseBasicParsing -OutFile
$Env:ProgramFiles\Docker\docker-compose.exe

# test the install by checking current version
PS> docker-compose version
```

Configure the Docker service by adding the following lines to file **C:\Windows\system32\drivers\etc\hosts**:

```
{
    "insecure-registries" : ["image-repo.ricms:5000"]
}
```

Then restart docker:

```
PS> Stop-Service docker
PS> Start-Service docker
```

4.3.5.2.2Join the Swarm

On the worker node, run the commands below to join the swarm cluster:

```
PS> docker swarm leave --force
PS> docker swarm join --token <SWMTKN> <HOST-KUBERNETES-MASTER-1>:2377
```

On the manager node, run the commands below to label and verify a worker node after joining it to the swarm:

```
$ sudo docker node ls
$ sudo docker node update --label-add nodeid=win2019 <NODE-ID>
$ sudo docker node update --label-add hcs=true <NODE-ID>
$ docker node ls -q | xargs docker node inspect -f '{{ .ID }} [{{
.Description.Hostname }}]: {{ .Spec.Labels }}'
```

4.3.6 Windows Authentication Servers

4.3.6.1 Windows Authentication Server

Add the following line to C:\Windows\system32\drivers\etc\hosts file:

```
<IMAGE-REPO-IP> image-repo.ricms
```

Install Docker EE for Windows following the instructions for Swarm Windows worker nodes in section 4.3.4.2.1.

Download, verify the checksum, and install the ASP.NET Core 2.1 Runtime - Windows Hosting Bundle:

> https://dotnet.microsoft.com/download/dotnet-core/2.1
>
> At the time of this writing, v2.1.14 was installed, however this should match the dotnet core version use to build the project code.

In preparation to install the WindowsApiCaller as a Windows service, download and install .NET Core 2.1.13 runtime libraries.

> https://download.visualstudio.microsoft.com/download/pr/d046f80d-8ad4-4bb9-8db6-8510105de979/07319c666f9951e15c607aed260ab12d/dotnet-runtime-2.1.13-win-x64.exe
>
> https://download.visualstudio.microsoft.com/download/pr/69d3ca05-a3f7-493c-816d-4b6ff0d9adeb/52de650ab7f96968e2718c418ac3d206/aspnetcore-runtime-2.1.13-win-x64.exe

4.3.6.2 Active Directory Domain Server

At most organizations, Active Directory (AD) hosts will be pre-configured by an Information Technology and/or Security team. In order to create groups and users, you may need to request the actions be performed be the appropriate IT/Security team.

Install the following as VM:

```
OS Name:                  Microsoft Windows Server 2019 Datacenter
OS Version:               10.0.17763 N/A Build 17763
Virtual Processor(s):     1
Total Physical Memory:    8,192 MB
```

During the installation select the following features:

- Active Directory Domain Services
- DNS Server

Enter the name for the network Domain, such as <AD-DOMAIN>.

Restart the server.

Log into the server as administrator.

Set the AD password restrictions as follows:

- Not contain the user's account name or parts of the user's full name that exceed two consecutive characters

- Be at least eight characters in length
- Contain characters from three of the following four categories:
- English uppercase characters (A through Z)
- English lowercase characters (a through z)
- Base 10 digits (0 through 9)
- Non-alphabetic characters (for example,!, $, #, %)
- Complexity requirements are enforced when passwords are changed or created.
- Account Lockout Policy:

o Account lockout duration	30 minutes
o Account lockout threshold	5 invalid logon attempts
o Reset account lockout counter after	30 minutes

Account Policies

1. Open the Server Manager from the Start menu.
2. Click on Tools, and then click on "Group Policy Management".
3. Right-click of "Default Domain Policy" which it will be located under your domain name and click Edit.
4. Expand the policy Computer Configuration -> Policies -> Windows Settings -> Security Settings -> Account Policies -> Password Policy
5. From the right pane do a double-click on the policy named "**Password Must Meet Complexity Requirements**".
6. Check "Define this policy setting", and selected Enabled.
7. Click OK
8. Double Click on the "Enforce password history" on the right panel and select the "Security Policy Settings" tab. Enter the above values in the appropriate location. Once completed, click Apply. You can review your policy by going to the "Explain" tab. Click OK to close the dialog.
9. Repeat for "Maximum password age", "Minimum password age", and "Minimum password length".
10. Select **Account Lockout Policy** on the left panel.
11. Select the "**Account lockout duration**", right click and select Edit. Enter the above value in the appropriate location. Once completed, click OK to close the dialog.
12. Do the same for "Account lockout threshold", and "Reset account lockout counter after".

Create AD Users and Groups

Active Directory groups are used to restrict access for different areas of the website. Request the following groups to be created by your IT/Security Team (the 'Group Name' does not have to be the actual Active Directory name):

Table 6 - Active Directory Groups

Group Name	Purpose
RICMS Admin	Admin user for developers, has all permissions and is apart of all Device Groups.
RICMS User	General purpose group that has only a few base level permissions.
Signal Operator	Has base level SOT permission.
Signal Approver	Can approve SOT related devices.
Signal Signer	Can manage SOT signed plans.
RPS User	Has base level RPS permissions.
RPS Admin	Admin group for all RPS related features.
RPS Device Approver	Can approve RPS related devices.
Event User	Can view and edit events.
Agency 1 Device Group	Device Group for Agency 1.
Agency 2 Device Group	Device Group for Agency 2.
Agency 3 Device Group	Device Group for Agency 3.

Once these groups are created, create these temporary users for testing purposes (the ‘Username’ does not have to be the actual Active Directory username):

Table 7 - Active Directory Users

Username	Member of
RpsAdmin	RICMS User, RPS Admin, Agency 3 Device Group
Agency1	RICMS User, RPS Device Approver, Agency 1 Device Group
Agency2	RICMS User, RPS Device Approver, Agency 2 Device Group
SotUser	RICMS User, Signal Operator, Signal Approver, Agency 1 Device Group

SotAdmin	RICMS User, Signal Operator, Signal Approver, Signal Signer, Agency 1 Device Group, Agency 2 Device Group, Agency 3 Device Group
SwriUser1	RICMS User, RICMS Admin, Agency 1 Device Group, Agency 2 Device Group, Agency 3 Device Group
SwriUser2	RICMS User, RICMS Admin, Agency 1 Device Group, Agency 2 Device Group, Agency 3 Device Group
SwriUser3	RICMS User, RICMS Admin, Agency 1 Device Group, Agency 2 Device Group, Agency 3 Device Group
EPIC Admin	RICMS User, RICMS Admin, Agency 1 Device Group, Agency 2 Device Group, Agency 3 Device Group
EpicUser1	RICMS User, Event User
EpicUser2	RICMS User, Event User
EpicUser3	RICMS User

Request a service account to be created with read access to be used by the auth services.

1. In the left navigation pane, right-click **Managed Service Accounts** and select **New User**.
2. Enter the First Name, Last Name, and “**sa_ricms**” as the logon name of the user and click Next. The account name may be provided by your IT/Security team.
3. Enter the **Password** and **Confirm Password**, then click Next and Finish.

Add Windows computers to Domain

On Swarm Windows servers and Windows Authentication Server, log in as administrator. This step may be performed by your IT/Security team.

1. On the **Start** screen, type **Control Panel**, and then press ENTER.
2. Navigate to **System and Security**, and then click **System**.
3. Under Computer name, domain, and workgroup settings, click Change settings.
4. On the **Computer Name** tab, click **Change**.
5. Under **Member of**, click Domain, type ‘epicgrpinc.com’ as the name of the domain that this computer will join, and then click **OK**.
6. Click **OK**, and then restart the computer.

4.3.7 Power BI Report Servers

Install the following as VM:

Host Name:	***<HOST-POWER-BI>***
IP Address:	***<IP-POWER-BI>***
OS Name:	***Microsoft Windows Server 2019 Datacenter***
OS Version:	***10.0.17763 N/A Build 17763***
Virtual Processor(s):	***1***
Total Physical Memory:	***16,384 MB***

4.4 Database Installation

4.4.1 MongoDB

MongoDB installation instructions are documented on the MongoDB website and should be referenced at the time of installation (latest changes are updated on the company's website) at the following location:

https://docs.mongodb.com/manual/tutorial/install-mongodb-enterprise-on-ubuntu/

Follow the instructions section entitled "Using .deb packages (recommended)".

The instructions to install/initialize a replica set (for high availability) are documented separately at the following location:

https://docs.mongodb.com/manual/tutorial/deploy-replica-set-with-keyfile-access-control/#deploy-new-replica-set-with-keyfile-access-control

4.4.1.1 Install MongoDB Enterprise

Perform the following steps on <HOST-MONGODB-0>, <HOST-MONGODB-1>, and <HOST-MONGODB-2> to install a MongoDB Enterprise Server 3 node replica set:

1. Add MongoDB pgp key

```
sudo wget -qO - https://www.mongodb.org/static/pgp/server-4.2.asc |
sudo apt-key add –
```

2. Add the MongoDB repository

```
sudo echo "deb [ arch=amd64 ] http://repo.mongodb.com/apt/ubuntu
bionic/mongodb-enterprise/4.2 multiverse" | sudo tee
/etc/apt/sources.list.d/mongodb-enterprise.list
```

3. Update package database

```
sudo apt-get update
```

4. Install MongoDB Enterprise 4.2.1 software

```
sudo apt-get install -y mongodb-enterprise=4.2.1 mongodb-enterprise-
server=4.2.1 mongodb-enterprise-shell=4.2.1 mongodb-enterprise-
mongos=4.2.1 mongodb-enterprise-tools=4.2.1
```

4.4.1.2 Setup MongoDB Enterprise

Perform the following steps on <HOST-MONGODB-0>, <HOST-MONGODB-1>, and <HOST-MONGODB-2>:

1. Ensure mongod service is stopped

   ```
   sudo service mongod stop
   ```

2. Update /etc/mongod.conf with the following

   ```
   storage:
     dbPath: /data/mongo/db
     journal:
       enabled: true

   systemLog:
     destination: file
     logAppend: true
     path: /data/mongo/log/mongod.log

   net:
     port: 27017
     bindIpAll: true

   processManagement:
     timeZoneInfo: /usr/share/zoneinfo

   security:
     keyFile: /home/mongodb/keyfile

   replication:
     replSetName: ricms
   ```

3. Create a directory for the cluster keyfile

   ```
   sudo mkdir /home/mongodb
   ```

4. Change ownership of the keyfile directory

   ```
   sudo chown mongodb:mongodb -R /home/mongodb
   ```

5. Create the data and log directories

   ```
   sudo mkdir /data/mongo/db /data/mongo/log
   ```

6. Change ownership of the data and log directories

   ```
   sudo chown -R mongodb:mongodb /data/mongo
   ```

Perform the following **only** on <HOST-MONGODB-0>:

1. Create the cluster key

   ```
   sudo openssl rand -base64 756 > /home/mongodb/keyfile
   ```

2. Change permission of keyfile

   ```
   sudo chmod 400 /home/mongodb/keyfile
   ```

Perform the following on <HOST-MONGODB-1> and <HOST-MONGODB-2>:

1. Create the keyfile

   ```
   sudo touch /home/mongodb/keyfile
   ```

2. Edit the /home/mongodb/keyfile and copy/paste the content from /home/mongodb/keyfile on <HOST-MONGODB-0> and save the file

3. Change permission of the keyfile
   ```
   sudo chmod 400 /home/mongodb/keyfile
   ```

Start mongod on all 3 hosts

```
sudo service mongod start
```

Initiate the replicate set

1. ssh to <HOST-MONGODB-0>
   ```
   ssh <USERNAME-AD-HUMAN-ACCOUNT>@<HOST-MONGODB-0>
   ```
2. Open mongo shell
   ```
   mongo
   ```
3. Copy/paste the following into the mongo shell
   ```
   rs.initiate( {
      _id : "ricms",
      members: [
         { _id: 0, host: "<HOST-MONGODB-0>:27017" },
         { _id: 1, host: "<HOST-MONGODB-1>:27017" },
         { _id: 2, host: "<HOST-MONGODB-2>:27017" }
      ]
   })
   ```
4. Frequently check whether initialization is complete
   a. Check the status
      ```
      rs.status().members
      ```
   b. When initialization is complete one member will have a stateStr value of PRIMARY
   c. Make note of the PRIMARY member's name
5. Exit the mongo shell
   ```
   exit
   ```

4.4.1.3 Add RICMS users, indexes, and views

Perform the following on the PRIMARY replica set member:

1. Open mongo shell
   ```
   mongo
   ```
2. Copy/paste the following to create a root user name admin
   ```
   db.getSiblingDB("admin").createUser(
     {
       user: "admin",
       pwd: "<PASSWORD-MONGODB-ADMIN>",
       roles: [ { role: "root", db: "admin" }]
     }
   )
   ```
3. Copy/paste the following to create the ricms_dfe_read_only user
   ```
   db.getSiblingDB("admin").createUser(
     {
       user: "ricms_dfe_read_only",
       pwd: "<PASSWORD-MONGODB-RICMS-DFE-READ-ONLY>",
       roles: [
   ```

```
  { role: "read", db: "sunguide" },
  { role: "read", db: "itsiqa" },
  { role: "read", db: "signal_controller_log" },
  { role: "read", db: "static_data" },
  { role: "read", db: "siia" },
  { role: "readWrite", db: "transit" },
  { role: "readWrite", db: "nws" }
  ]
 }
)
```

4. Copy/paste the following to create the ricms_dfe_read_write_user

```
db.getSiblingDB("admin").createUser(
 {
  user: "ricms_dfe_read_write",
  pwd: "<PASSWORD-MONGODB-RICMS-DFE-READ-WRITE>",
  roles: [
  { role: "readWrite", db: "sunguide" },
  { role: "readWrite", db: "itsiqa" },
  { role: "readWrite", db: "signal_controller_log" },
  { role: "readWrite", db: "static_data" },
  { role: "readWrite", db: "siia" },
  { role: "read", db: "transit" },
  { role: "read", db: "nws" }
  ]
 }
)
```

5. Exit the mongo shell

```
exit
```

6. Copy the files from RICMS\Deploy\MongoDB (from RICMS source repository) to the PRIMARY server's /home/mongodb directory
7. Create the views and indexes

```
mongo "mongodb://admin:<PASSWORD-MONGODB-ADMIN>@<HOST-MONGODB-0>:27017,
<HOST-MONGODB-1>:27017,<HOST-MONGODB-
2>:27017/admin?readPreference=primary&connectTimeoutMS=10000&authSource
=admin&authMechanism=SCRAM-SHA-1" create_indexes.js create_views.js
```

4.4.2 *SQL Server*

This section does not replace the detailed instructions published by Microsoft for the installation of SQL Server 2016. These instructions only describe the highlights and are intended for persons experienced with installing SQL Server products. It is helpful if all previous SQL Server installations are removed. Remove empty SQL Server directories from previous installations.

4.4.2.1 Install SQL Server

General Installation for SQL Server are provided at this URL:

https://docs.microsoft.com/en-us/sql/database-engine/install-windows/install-sql-server?view=sql-server-2016

- Review installation requirements, system configuration checks, and security considerations for a SQL Server installation.
- Run SQL Server Setup to install or upgrade to a later version.
- Use SQL Server utilities to configure SQL Server.

4.4.2.2 Create Databases and Logins

To create the SQL Server databases, logins, and users for the RICMS system, run the two scripts below. Each script includes instructions for usage and required privileges:

- Deploy/SqlServer/sysadmin-login.sql
- Deploy/SqlServer/database-logins-users.sql

4.4.3 *GIS Databases*

The steps to install GIS Database are included in the ArcGIS Enterprise Application installation Section 4.5.3.1

4.5 Application Installation

4.5.1 Cloudera Installation

Cloudera installation involves two phases:

- Nodes setup with hardware configuration and OS installation, Section 4.3.1 Cloudera Nodes Preparation
- Cloudera software installation and configuration of nodes, Section 4.5.1.1 Cloudera Software Installation

4.5.1.1 Cloudera Software Installation

Install Cloudera software components and configure a repository using package management tool apt.

4.5.1.1.1 Configure Cloudera Manager Repository

Perform below steps on Cloudera utility node as administrator to setup repository:

1. Navigate to the below URL in a browser and download the ***cloudera-manager.list*** file for the OS (ubuntu1804 in our case).

https://<CLOUDERA-REPO-USERNAME>:<CLOUDERA-REPO-PASSWORD>@archive.cloudera.com/p/cm6/6.3.3/ubuntu1804/apt

2. Save the file ***cloudera-manager.list*** to the path ***/etc/apt/sources.list.d/*** directory on the Cloudera Manager Server host.

3. Modify the contents of the file cloudera-manager.list to add username:password to the url and to switch to https instead of http.

   Existing contents of the file:

```
# Cloudera Manager 6.3.3
deb [arch=amd64] http://archive.cloudera.com/p/cm6/6.3.3/ubuntu1804/apt bionic-cm6.3.3 contrib
```

   Modified contents of the file:

```
# Cloudera Manager 6.3.3
deb [arch=amd64] https://<CLOUDERA-REPO-USERNAME>:<CLOUDERA-REPO-PASSWORD>@archive.cloudera.com/p/cm6/6.3.3/ubuntu1804/apt bionic-cm6.3.3 contrib
```

4. Import the repository signing GPG key

```
$ wget https://<CLOUDERA-REPO-USERNAME>:<CLOUDERA-REPO-PASSWORD>@archive.cloudera.com/p/cm6/6.3.3/ubuntu1804/apt/archive.key
$ sudo apt-key add archive.key
```

4.5.1.1.2 Enable password less ssh login to administrator

Ensure that all the Cloudera Cluster nodes has administrator user created and then configure password-less ssh login from Cloudera manager node to start the services automatically post installation.

Run the following command on each node if administrator user doesn't exist.

```
$ sudo adduser administrator
$ sudo usermod –Ag sudo administrator
```

Run the below commands on Cloudera Manager.

```
ssh-keygen -t rsa
ssh-copy-id administrator@<remote host> # for each node on the cluster
```

4.5.1.1.3 Install JDK

Run the below commands on all hosts

1. Update the packages

```
$ sudo apt-get update
```

2. Install oracle JDK

```
$ sudo apt-get install oracle-j2sdk1.8
```

4.5.1.1.4 Install Cloudera Manager Server

To install Cloudera Manager Server, use the following steps:

1. Run the below command on <HOST-CLOUDERA-UTILITY0> to install Cloudera Manager Server Packages

   ```
   $ sudo apt-get install -y cloudera-manager-daemons cloudera-manager-agent cloudera-manager-server
   ```

2. Enable Auto-TLS. This command enables Auto-TLS using internal CA.

   ```
   sudo JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64 /opt/cloudera/cm-agent/bin/certmanager setup --configure-services
   ```

4.5.1.1.5 Install & Configure Cloudera Databases

To install and configure MariaDB database[1] on cloudera manager use the following steps:

1. Run the following command to install MariaDB Server:

   ```
   $ sudo apt-get install –y mariadb-server
   ```

2. Stop the MariaDB Server to configure it.

   ```
   $ sudo service mariadb stop
   ```

3. If they exist, move old InnoDB log files to a backup location.

   ```
   $ cd /var/lib/mysql/
   $ sudo mkdir SQL_Backup_Location
   $ sudo mv ib* SQL_Backup_Location/
   ```

4. Take backup of existing configuration file and edit the file.

   ```
   $ cd /etc/mysql/
   $ sudo cp my.cnf my.cnf_Original
   $ sudo vi my.cnf
   ```

   Replace the contents of the file with below data

   ```
   # The MariaDB configuration file
   #
   # The MariaDB/MySQL tools read configuration files in the following order:
   # 1. "/etc/mysql/mariadb.cnf" (this file) to set global defaults,
   # 2. "/etc/mysql/conf.d/*.cnf" to set global options.
   # 3. "/etc/mysql/mariadb.conf.d/*.cnf" to set MariaDB-only options.
   # 4. "~/.my.cnf" to set user-specific options.
   #
   # If the same option is defined multiple times, the last one will apply.
   #
   ```

[1] Mariadb is a fork of mysql; the installation instructions correctly reference mysql.

```
# One can use all long options that the program supports.
# Run program with --help to get a list of available options and with
# --print-defaults to see which it would actually understand and use.

#
# This group is read both both by the client and the server
# use it for options that affect everything
#
[client-server]
[mysqld]

datadir=/var/lib/mysql
socket=/var/run/mysqld/mysqld.sock
transaction-isolation = READ-COMMITTED
# Disabling symbolic-links is recommended to prevent assorted security risks;
# to do so, uncomment this line:
symbolic-links = 0
# Settings user and group are ignored when systemd is used.
# If you need to run mysqld under a different user or group,
# customize your systemd unit file for mariadb according to the
# instructions in http://fedoraproject.org/wiki/Systemd

key_buffer = 16M
key_buffer_size = 32M
max_allowed_packet = 32M
thread_stack = 256K
thread_cache_size = 64
query_cache_limit = 8M
query_cache_size = 64M
query_cache_type = 1

max_connections = 550
#expire_logs_days = 10
#max_binlog_size = 100M

#log_bin should be on a disk with enough free space.
#Replace '/var/lib/mysql/mysql_binary_log' with an appropriate path for your
#system and chown the specified folder to the mysql user.
log_bin=/var/lib/mysql/mysql_binary_log
```

```
#In later versions of MariaDB, if you enable the binary log and do not set
#a server_id, MariaDB will not start. The server_id must be unique within
#the replicating group.
server_id=1

binlog_format = mixed

read_buffer_size = 2M
read_rnd_buffer_size = 16M
sort_buffer_size = 8M
join_buffer_size = 8M

# InnoDB settings
innodb_file_per_table = 1
innodb_flush_log_at_trx_commit  = 2
innodb_log_buffer_size = 64M
innodb_buffer_pool_size = 4G
innodb_thread_concurrency = 8
innodb_flush_method = O_DIRECT
innodb_log_file_size = 512M

[mysqld_safe]
log-error=/var/log/mariadb/mariadb.log
pid-file=/var/run/mariadb/mariadb.pid

#
# include all files from the config directory
#
# Import all .cnf files from configuration directory
#!includedir /etc/mysql/conf.d/
#!includedir /etc/mysql/mariadb.conf.d/
bind-address = 0.0.0.0
```

5. Create directory required for MariaDB and set ownership

```
$ sudo mkdir /run/mariadb
$ sudo chown mysql:mysql /run/mariadb/
```

6. Start mariadb and enable start of the service on boot

```
$ sudo service mysql start
$ sudo service mysql status
$ sudo systemctl enable mariadb.service
```

7. Run the following command to initialize the database and set root password for the database

   ```
   $ sudo /usr/bin/mysql_secure_installation
   ```

   a. It will prompt for the current password. Press enter since there is not one currently existing.
   b. Next, it will prompt: Set root password? [Y/N]
      i. Type Y and press enter
   c. It will prompt the user to enter a new password <MARIADB-PASSWORD> twice and will display a success message after completing. It will prompt to remove anonymous users [Y/N]
      i. Type Y and press enter
   d. Next, it will prompt: Disallow root login remotely? [Y/N]
      i. If you don't want to login from a remote machine with root, the type Y and press enter.
      ii. If you want to login from a remote machine with root, they N and press enter.
   e. Next, it will prompt: Remove test database and access to it? [Y/N]
      i. Type Y and press enter
   f. Next, it will prompt: Reload privilege tables now? [Y/N]
      i. Type Y and press enter

4.5.1.1.6 Create Databases

Create databases and service accounts for components that require databases:

- Cloudera Manager Server
- Cloudera Management Service roles:
  - Activity Monitor
  - Reports Manager

Record the values you enter for database names, usernames, and passwords. The Cloudera Manager Installation wizard requires this information to correctly connect to these databases.

1. Log in as the root user, or another user with privileges to create database, users and grants

   ```
   $ mysql -u root –p
   Enter password:<MARIADB-PASSWORD>
   ```

2. Create databases and users for each service:

   ```
   MariaDB [(none)]> CREATE DATABASE scm DEFAULT CHARACTER SET utf8 DEFAULT COLLATE utf8_general_ci;
   ```

```
MariaDB [(none)]> GRANT ALL ON scm.* TO 'scm'@'%' IDENTIFIED BY '<MARIADB-PASSWORD>';
MariaDB [(none)]> CREATE DATABASE amon DEFAULT CHARACTER SET utf8 DEFAULT COLLATE utf8_general_ci;
MariaDB [(none)]> GRANT ALL ON amon.* TO 'amon'@'%' IDENTIFIED BY '<MARIADB-PASSWORD>';
MariaDB [(none)]> CREATE DATABASE rman DEFAULT CHARACTER SET utf8 DEFAULT COLLATE utf8_general_ci;
MariaDB [(none)]> GRANT ALL ON rman.* TO 'rman'@'%' IDENTIFIED BY '<MARIADB-PASSWORD>';
MariaDB [(none)]> CREATE DATABASE hue DEFAULT CHARACTER SET utf8 DEFAULT COLLATE utf8_general_ci;
MariaDB [(none)]> GRANT ALL ON hue.* TO 'hue'@'%' IDENTIFIED BY '<MARIADB-PASSWORD>';
MariaDB [(none)]> CREATE DATABASE metastore DEFAULT CHARACTER SET utf8 DEFAULT COLLATE utf8_general_ci;
MariaDB [(none)]> GRANT ALL ON metastore.* TO 'hive'@'%' IDENTIFIED BY '<MARIADB-PASSWORD>';
MariaDB [(none)]> CREATE DATABASE sentry DEFAULT CHARACTER SET utf8 DEFAULT COLLATE utf8_general_ci;
MariaDB [(none)]> GRANT ALL ON sentry.* TO 'sentry'@'%' IDENTIFIED BY '<MARIADB-PASSWORD>';
MariaDB [(none)]> CREATE DATABASE nav DEFAULT CHARACTER SET utf8 DEFAULT COLLATE utf8_general_ci;
MariaDB [(none)]> GRANT ALL ON nav.* TO 'nav'@'%' IDENTIFIED BY '<MARIADB-PASSWORD>';
MariaDB [(none)]> CREATE DATABASE navms DEFAULT CHARACTER SET utf8 DEFAULT COLLATE utf8_general_ci;
MariaDB [(none)]> GRANT ALL ON navms.* TO 'navms'@'%' IDENTIFIED BY '<MARIADB-PASSWORD>';
MariaDB [(none)]> CREATE DATABASE oozie DEFAULT CHARACTER SET utf8 DEFAULT COLLATE utf8_general_ci;
MariaDB [(none)]> GRANT ALL ON oozie.* TO 'oozie'@'%' IDENTIFIED BY '<MARIADB-PASSWORD>';
```

Note: Record the password you chose; it will be required when you setup the Cloudera Manager.

3. Confirm that you have created all of the databases:

```
MariaDB [(none)]> SHOW DATABASES;
```

You can also confirm the privilege grants for a given user by running:

```
MariaDB [(none)]> SHOW GRANTS FOR '<user>'@'%';
```

4.5.1.1.7 Setup Cloudera Manager Database

Cloudera Manager Server includes a script that can create and configure a database for itself.

1. To access the script and set up the Cloudera Manager SCM Database run the following command:

   ```
   $ sudo /opt/cloudera/cm/schema/scm_prepare_database.sh mysql scm scm
   ```

   When prompted provide the password chosen in prior step

4.5.1.1.8 Start Cloudera Manager Server

1. Start Cloudera Manager Server:

   ```
   $ sudo systemctl start cloudera-scm-server
   ```

2. Wait several minutes for the Cloudera Manager Server to start. To observe the startup process, run the following on the Cloudera Manager Server host:

   ```
   $ sudo tail –f /var/log/cloudera-scm-server/cloudera-scm-server.log
   ```

When the following entry is visible in the log, the Cloudera Manager Admin Console is ready:

```
INFO WebServerImpl:com.cloudera.server.cmf.WebServerImpl: Started Jetty server.
```

4.5.1.1.9Install CDH and Other Software

Follow steps below to Install and configure CDH and other CDH services through the Cloudera Manager web UI.

1. In a web browser go to https://< HOST-CLOUDERA-UTILITY-FQDN-0>:7183
   a. If a security warning is displayed either (depending on browser) choose to trust the certificate or proceed.
   b. Add < HOST-CLOUDERA-UTILITY-FQDN-0> to your local hosts file to facilitate name resolution or replace it with the corresponding IP in the URL.
2. Log in to Cloudera Manager Admin Console. The default credentials are:
   a. Username: admin
   b. Password: admin

Note: the password can be changed after running the installation wizard

The subsequent follow through installation instructions of CDH software are available at the following location:

https://docs.cloudera.com/documentation/enterprise/latest/topics/install_software_cm_wizard.html#cm_installation_wizard

Complete the installation and cluster setup using the install wizard and the instructions mentioned at above web site.

3. Click continue on the welcome page
4. The End User License Terms and Conditions page is displayed. Read the terms, conditions, and then check the box labeled “Yes, I accept the End User License Terms and Conditions” to accept them. Click Continue and the installation wizard launches.
5. Choose Cloudera Enterprise as edition
6. Install the License

   a. Click the Select License File text field.
   b. Browse to the location of your license file, select the file, and then click Open.
   c. Click the Upload button.

7. Click continue to proceed to next step.
8. View the Welcome page and press Continue.
9. In Cluster Basics page, specify the cluster name and click continue to proceed.
10. View the Setup Auto-TLS page and press Continue. There is nothing to configure on this page because the certificate manager was set up for Auto-TLS when installing Cloudera Manager in an earlier step.
11. On the Specify Hosts page, to enable Cloudera Manager to automatically discover hosts on which to install CDH and managed services, enter the following list of cluster hostnames in Specify Hosts page:
    a. <HOST-CLOUDERA-UTILITY-FQDN-0>, <HOST-CLOUDERA-EDGE-FQDN-0>, <HOST-CLOUDERA-MASTER-FQDN-0>, <HOST-CLOUDERA-MASTER-FQDN-1>, <HOST-CLOUDERA-KAFKA-FQDN-0>, <HOST-CLOUDERA-KAFKA-FQDN-1>, <HOST-CLOUDERA-WORKER-FQDN-0>, <HOST-CLOUDERA-WORKER-FQDN-1>, <HOST-CLOUDERA-WORKER-FQDN-2>, <HOST-CLOUDERA-WORKER-FQDN-3>
    b. Click Search.
    c. Verify that the number of hosts shown matches the number of hosts where you want to install services.
12. Click continue to proceed to next step.
13. On the Select Repository page, in the Cloudera Manager Agent section, select Public Cloudera Repository for the Cloudera Manager Agent software.
14. In the CDH and other software section, select to use Parcels as the repository method.
15. Select Required CDH version and click continue to next step.
16. Accept JDK License to continue to next step.
17. On the Enter Login Credentials page, select root as the account with All hosts accept same password as the authentication method and provide password for root account created in the Section 4.3.1.2.2.11. Use the default for number of simultaneous installations and click continue to proceed to next step.
18. Cloudera Manager will now install the agents on all nodes of the cluster. This may take a few minutes.
19. The Install Parcels page will display. Press Continue when the button becomes active (i.e., after the parcels have been downloaded and installed to all hosts)
20. Inspect Cluster and click continue.
21. This completes the Cluster installation wizard and launches Add Cluster - Configuration wizard.

4.5.1.1.10 Set Up a Cluster

1. On the Select Services page, select “All Services (Cloudera Enterprise Data Hub)” as the service to install.
2. Check the box Include Cloudera Navigator and click continue.
3. On the Assign Roles page, select “Roles” that are assigned to each node according to the following figure and press continue.

HBase			
Master	HBase REST Server	HBase Thrift Server	RegionServer
<HOST-CLOUDERA-EDGE-FQDN-0>	<HOST-CLOUDERA-UTILITY-FQDN-0>	<HOST-CLOUDERA-UTILITY-FQDN-0>	Same As DataNode
HDFS			
NameNode	SecondaryNameNode	Balancer	HttpFS
<HOST-CLOUDERA-MASTER-FQDN-0>	<HOST-CLOUDERA-MASTER-FQDN-1>	<HOST-CLOUDERA-EDGE-FQDN-0>	<HOST-CLOUDERA-EDGE-FQDN-0>
NFS Gateway	DataNode		
<HOST-CLOUDERA-EDGE-FQDN-0>	<HOST-CLOUDERA-WORKER-FQDN- [0-3]>		
Hive			
Gateway	Hive Metastore Server	WebHCat Server	HiveServer2
All Nodes	<HOST-CLOUDERA-UTILITY-FQDN-0>	<HOST-CLOUDERA-EDGE-FQDN-0>	<HOST-CLOUDERA-EDGE-FQDN-0>
Hue			
Hue Server	Load Balancer		
<HOST-CLOUDERA-EDGE-FQDN-0>	<HOST-CLOUDERA-EDGE-FQDN-0>		
Impala			
Impala StateStore	Impala Catalog Server	Impala Daemon	
<HOST-CLOUDERA-UTILITY-FQDN-0>	<HOST-CLOUDERA-UTILITY-FQDN-0>	Same As DataNode	
Key-Value Store Indexer			
Lily HBase Indexer			
<HOST-CLOUDERA-EDGE-FQDN-0>			

Cloudera Management Service			
Service Monitor <HOST-CLOUDERA-EDGE-FQDN-0>	Activity Monitor <HOST-CLOUDERA-EDGE-FQDN-0>	Host Monitor <HOST-CLOUDERA-EDGE-FQDN-0>	Reports Manager <HOST-CLOUDERA-EDGE-FQDN-0>
Event Server <HOST-CLOUDERA-EDGE-FQDN-0>	Alert Publisher <HOST-CLOUDERA-EDGE-FQDN-0>	Navigator Audit Server <HOST-CLOUDERA-EDGE-FQDN-0>	Navigator Metadata Server <HOST-CLOUDERA-EDGE-FQDN-0>
Telemetry Publisher <HOST-CLOUDERA-EDGE-FQDN-0>			
Oozie			
Oozie Server <HOST-CLOUDERA-UTILITY-FQDN-0>			
Solr			
Solr Server <HOST-CLOUDERA-EDGE-FQDN-0>			
Spark			
History Server <HOST-CLOUDERA-EDGE-FQDN-0>	Gateway <HOST-CLOUDERA-EDGE-FQDN-0>		
YARN (MR2 Included)			
ResourceManager <HOST-CLOUDERA-MASTER-FQDN-0>	JobHistory Server <HOST-CLOUDERA-MASTER-FQDN-0>	Node Manager Same As DataNode	
ZooKeeper			
Server <HOST-CLOUDERA-MASTER-FQDN- [0-1]> <HOST-CLOUDERA-UTILITY-FQDN-0>			

4. In the next step, on Setup Database page, specify the information of the databases created in the Section 4.5.1.2.6 according to the following figure and press continue.

Hive					
Type MySQL	Use JDBC URL Override No	Database Hostname <HOST-CLOUDERA-UTILITY-0>	Database Name Metastore	Username Hive	Password <MARIADB-PASSWORD>
Activity Monitor					
Type MySQL	Database Hostname <HOST-CLOUDERA-UTILITY-0>	Database Name amon	Username amon	Password <MARIADB-PASSWORD>	
Reports Manager					
Type MySQL	Database Hostname <HOST-CLOUDERA-UTILITY-0>	Database Name rman	Username rman	Password <MARIADB-PASSWORD>	
Navigator Audit Server					
Type MySQL	Database Hostname <HOST-CLOUDERA-UTILITY-0>	Database Name nav	Username nav	Password <MARIADB-PASSWORD>	
Navigator Metadata Server					
Type MySQL	Database Hostname <HOST-CLOUDERA-UTILITY-0>	Database Name navms	Username navms	Password <MARIADB-PASSWORD>	
Oozie Server					
Type MySQL	Database Hostname	Database Name oozie	Username oozie	Password <MARIADB-PASSWORD>	

	<HOST-CLOUDERA-UTILITY-0>			
Hue				
Type	Database Hostname	Database Name	Username	Password
MySQL	<HOST-CLOUDERA-UTILITY-0>	hue	hue	<MARIADB-PASSWORD>

5. On the Review Changes page, review the configuration parameters and click Continue.
6. On the Command Details page confirm that the services started successfully and press Continue.
7. On the Summary page, click Finish to complete the Wizard after everything is successfully installed.
8. The browser displays the Cloudera Manager home page.

4.5.1.1.11 Add the Kafka service

1. From the Home page, choose Add Service from the pulldown menu to the right of Cluster Name.
2. On the Add Service to Cluster page, choose Kafka and press Continue.
3. On Assign Roles for Kafka page, select hosts as listed below and click Continue.
   a. Kafka Brokers <HOST-CLOUDERA-KAFKA-FQDN-0>, <HOST-CLOUDERA-KAFKA-FQDN-1> & <HOST-CLOUDERA-WORKER-FQDN-0>
   b. Mirror Maker: None
   c. Gateway: None
4. On the Review Changes page, set source and destination broker list for mirrormaker even though we don’t want to run mirrormaker. Provide the <HOST-CLOUDERA-KAFKA-IP-0>:9092 just to get past this step. Once service installation is done find and delete mirrormaker role.
5. Proceed through the wizard to the Command Details page. On the Command Details page, verify that the Kafka service was started successfully and press Continue.
6. Configure Kafka brokers to listen for insecure clients
   a. From Cloudera Manager Home page, click on Kafka service.
   b. From the Kafka service page, press Configuration.
   c. In the search box, type kafka.properties to display the advanced configuration(safety valve) settings.
   d. For each broker <HOST-CLOUDERA-KAFKA-FQDN-[0-1]> and <HOST-CLOUDERA-WORKER-FQDN-0>, in the Kafka Broker (broker) box, copy/paste the following:

      listeners=PLAINTEXT://<broker>:9092,SSL:// <broker>:9093

      advertised.listeners=PLAINTEXT:// <broker>:9092,SSL:// <broker>:9093
   e. Press Save Changes
7. Set Kafka offsets.topic.replication.factor

   a. From the Kafka service Configuration page, type offsets.topic.replication.factor in the search box.
   b. Change the value of offsets.topic.repliacation.factor to 3 and press Save Changes.
8. Since changes were made to the Kafka configuration, Cloudera Manager will need to deploy the configuration change and restart the Kafka service. Choose this option when prompted.

4.5.1.1.12 Enable HDFS for high availability

1. From the Home page, click on the HDFS service.
2. On the HDFS service page, choose Enable High Availability from the Actions menu to launch the Enable High Availability for HDFS wizard.
3. On the Getting Started Page, enter the following and press Continue
   a. Nameservice Name: nameservice1
4. On the Assign Roles page, choose the following and press Continue
   a. NameNode Hosts:
      i. <HOST-CLOUDERA-MASTER-FQDN-0> (Current)
      ii. <HOST-CLOUDERA-MASTER-FQDN-1>
   b. Journal Node Hosts:
      i. <HOST-CLOUDERA-MASTER-FQDN-[0-1]>
      ii. <HOST-CLOUDERA-UTILITY-FQDN-0>
5. On the Review Changes page, ensure the following and press Continue
   a. JournalNode Edits Directory:
      i. <HOST-CLOUDERA-MASTER-FQDN-0>: /dfs/jn
      ii. <HOST-CLOUDERA-MASTER-FQDN-1>: /dfs/jn
      iii. <HOST-CLOUDERA-UTILITY-FQDN-0>: /dfs/jn
   b. Extra options:
      i. Check all 3 checkboxes
6. On the Command Details page, verify that High Availability and Automatic Failover was successfully enabled.
7. Update Hive Metastore NameNodes
   a. From the Cloudera Manager Home page, click on the Hive service.
   b. From the Hive service page, choose Stop from the Actions menu.
   c. After the service is stopped, choose Update Hive Metastore NameNodes from the Actions Menu.
   d. After the update completes, choose Start form the Actions menu.

4.5.1.1.13 Configure Data Directories

1. Before configuring Data Directories, create version-2 directory for zookeeper on <HOST-CLOUDERA-MASTER-FQDN-[0-1]>
   a. Create directory under /data03/zookeeper

```
$ mkdir version-2
```

   b. Change owner

   ```
   $ chown zookeeper.zookeeper version-2
   ```

2. Use the data directories.xlsx file to configure new values for all the data directories. From the Cloudera Manager Home page, click on Configuration and search for the Configuration Description or Configuration setting value in the search bar and fill/replace with the corresponding new values.
3. On <HOST-CLOUDERA-UTILITY-FQDN-0> change the directory of Mariadb data by creating a symbolic link to the path specified in the RICMS-VDD-3.1-Cloudera-Data-Directories.xlsx file.

4.5.1.2 Set up Auto-TLS using existing certificates

Cloudera recommends obtaining certificates from one of the trusted public/internal certificate authorities (CA) for TLS/SSL encryption for the cluster. Follow the below steps to set up Auto-TLS using certificates signed from trusted public/internal CA.

4.5.1.2.1Create Directory for Security Artifacts

Setting up Auto-TLS involves distributing the certificates, keys, truststore etc.,To keep things organized, it is recommended to create the directory.

```
$ sudo mkdir -p /opt/cloudera/security/pki
```

4.5.1.2.2Generate Server Key and CSR

1. Ensure that you have JAVA keytool on the machine that you are running the below commands.
2. Export JAVA_HOME

   ```
   $ export JAVA_HOME=/usr/java/jdk1.8.0_202
   ```

3. For each host in the Cloudera Cluster, run the below commands

   a. Generate Server Key:

   ```
   $ $JAVA_HOME/bin/keytool -genkeypair -alias <HOSTNAME> -keyalg RSA -keystore /opt/cloudera/security/pki/<HOSTNAME>.jks -keysize 2048 -dname "CN=<HOSTNAME>,OU=D5 ITS,O=Florida DOT,L=Deland,ST=FL,C=US" -ext san=dns:<HOSTNAME>
   ```

   Enter <CSR-PASSWORD> for storepass and keypass when prompted

   b. Generate CSR

   ```
   $ $JAVA_HOME/bin/keytool -certreq -alias <HOSTNAME> -keystore /opt/cloudera/security/pki/<HOSTNAME>.jks -file /opt/cloudera/security/pki/<HOSTNAME>.csr -ext san=dns:<HOSTNAME> -ext EKU=serverAuth,clientAuth
   ```

   Enter <CSR-PASSWORD> for storepass and keypass when prompted

4.5.1.2.3 Submit the CSR to the CA

1. Submit the CSR file to your certificate authority using the process and means required by the CA. For the certificate format, specify PEM (base64 ASCII)
2. The public CA will request specific details from you, to verify that you own the domain name contained in the CSR, before they issue the certificate.
3. When you receive the signed certificate from the CA, you can proceed with next step.

4.5.1.2.4 Verify the Certificate

1. From the public (or internal) CA, you receive the certificate for the server signed by the CA and several other digital artifacts, including root CA and possibly one (or more) intermediate CA certificates. Before distributing these to the hosts, make sure the certificate is in PEM format.
2. If the CA provided you with CSR certificates convert them to PEM files as shown below

```
$ openssl x509 –in <file>.cer -outform PEM - <file>.pem
```

3. The final PEM file should have the below format

```
-----BEGIN CERTIFICATE-----
<The encoded certificate is represented by multiple lines of exactly 64 characters, except
for the last line, which can contain 64 characters or fewer.>
-----END CERTIFICATE-----
```

4.5.1.2.5Import Keystore

Run the below commands for each Cloudera host to create host-key files.

1. Export from pks to p12

```
$ sudo keytool -importkeystore -srckeystore /opt/cloudera/security/pki/<HOSTNAME>.jks -destkeystore /opt/cloudera/security/pki/<HOSTNAME>-key.p12 -deststoretype PKCS12 -srcalias <HOSTNAME>
```

2. Extracting private key from p12 to pem

```
$ sudo openssl pkcs12 -in /opt/cloudera/security/pki/<HOSTNAME>-key.p12 -out /opt/cloudera/security/pki/<HOSTNAME>.pem
```

3. Renaming files to append .key

```
$ mv <HOSTNAME>.pem <HOSTNAME>.pem.key
```

4.5.1.2.6 Concatenate Root Certificate and Intermediate Certificate

Append the intermediate certificate to the root ca certificate with the below command

```
$ cat <root-ca-cert>.pem <intermediate-cert>.pem > ca-cert.pem
```

4.5.1.2.7 Generate and Distribute CM certificates

1. On <HOST-CLOUDERA-UTILITY-0> node, rename the folder /var/lib/cloudera-scm-server/certmanager if exists

```
$ mv /var/lib/cloudera-scm-server/certmanager/ /var/lib/cloudera-scm-server/certmanager_bkp_<date>/
```

2. Go to the below URL https://<HOST-CLOUDERA-UTILITY-FQDN-0>:7183/static/apidocs/ui/index.html
3. In the Swagger UI, navigate to ClouderaManagerResource -> /cm/commands/generateCmca
4. Under the generateCmca section, in the body section, paste the below code and click on Try it out!

```
{
"location": "/var/lib/cloudera-scm-server/certmanager",
"customCA": true,
"interpretAsFilenames": true,
"cmHostCert": <cloudera manager host certificate file path>",
"cmHostKey": "<cloudera manager host key file path>",
"caCert": "<ca-cert file path>",
"keystorePasswd": "<keystore password file path>",
"truststorePasswd": "<Truststore password file path>",
"hostCerts": [
{"hostname": "<host1 fully qualified name>", "certificate": "<host1 certificate file path>", "key": "<host1 key file path>"},
{"hostname": "<host2 fully qualified name>", "certificate": "<host2 certificate file path>", "key": "<host2 key file path>"}
<<Add required number of hosts using comma separated set enclosed in curly braces>>
],
"configureAllServices": "true",
"sshPort": 22,
"userName": "root",
"password": "<CLOUDERA-ROOT-PASSWORD> "
}
```

5. This runs commands to distribute the certificates and the Auto-TLS configuration is completed.
6. Verify the certificate expiry date to ensure that the Auto-TLS is configured properly.

4.5.1.3 Renewing Certificates

Follow the below steps in order to renew the certificates

1. Generate CSR as discussed in step 3b of section 4.5.1.2.2
2. Submit the CSR files to CA as discussed in section 4.5.1.2.3

3. Verify the certificate received from CA as discussed in section 4.5.1.2.4
4. Append intermediate certificate to root certificate as discussed in section 4.5.1.2.6
5. Generate and distribute CM certificates as discussed in 4.5.1.2.7

4.5.2 Elastic-stack Installation/Deployment

4.5.2.1 Elasticsearch

4.5.2.1.1 Creating a system user

A Linux system user is required to run the Elasticsearch service. This new user will not be given superuser privileges. The instructions in the remaining sections will denote which user is required to run the command. The following commands require superuser privileges.

1. Create a user
   ```
   $ sudo useradd elastic
   ```
2. Create a password for the new user
   ```
   $ sudo passwd elastic
   ```
3. Change the ownership of the `/data` sub directories to the new user.
   ```
   $ sudo chown elastic:elastic /data/*
   ```

4.5.2.1.2 Download Elasticsearch

1. As the elastic system user, download Elasticsearch
   ```
   $ curl -L -O
   https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-
   7.4.2-linux-x86_64.tar.gz
   $ tar -xvf elasticsearch-7.4.2-linux-x86_64.tar.gz
   ```

4.5.2.1.3Configure Elasticsearch

As the elastic system user, perform the following changes to the file `elasticsearch-7.4.2/config/elasticsearch.yml` on each node. The order of the following steps matches the order that they appear in the file.

1. Uncomment the `cluster.name` line and change the value to a desired cluster name. This value should be the same on all nodes.
2. Uncomment the `node.name` line and change the value to a desired node name. This value should be different on all nodes.
3. Uncomment the `path.data` line and remove the current value. The new value will be an array of file paths to the `/data` sub directories created in Elasticsearch Nodes Preparation. It will look like this:
   ```
   path.data:
   ```

```
- /data/00
- /data/01
.
.
.
- /data/XX
```

4. Uncomment the `network.host` line and change the value to `0.0.0.0`
5. Uncomment the `discovery.seed_hosts` line and remove the current values. The new values will be an array of all the Elasticsearch nodes' host names (including the current node).
6. Uncomment the `cluster.initial_master_nodes` line and remove the current values. The new values will be an array of all the node names created in the first step (including the current node).

As the elastic system user, perform the following changes to the file `elasticsearch-7.4.2/config/jvm.options` on all nodes.

1. Change the `-Xms1g` and `-Xmx1g` lines to the following:
   ```
   -Xms30g
   -Xmx30g
   ```

As a superuser, perform the following changes to the file `/etc/sysctl.conf` on all nodes.

1. Add the following line:
   ```
   vm.max_map_count=262144
   ```

4.5.2.1.4Configure an Elasticsearch systemd service

As a superuser, perform the following changes on all nodes.

1. Create a `.service` file
   ```
   $ sudo touch /etc/systemd/system/elasticsearch.service
   ```
2. Add the following lines to the newly created file:
   ```
   [Unit]
   Description=Elasticsearch server
   After=network.target

   [Service]
   Type=simple
   User=elastic
   Group=elastic
   WorkingDirectory=/home/elastic/elasticsearch-7.4.2
   ExecStart=/home/elastic/elasticsearch-7.4.2/bin/elasticsearch
   Restart=on-failure

   LimitNOFILE=65536
   ```

```
[Install]
WantedBy=multi-user.target
```

3. Enable/run the new systemd service:
```
$ sudo systemctl daemon-reload
$ sudo systemctl enable elasticsearch.service
$ sudo reboot
```
4. After performing the above step on all nodes, verify that the cluster is running and all nodes successfully joined the cluster.
   a. You can check the cluster status by running the following command:
   ```
   $ curl -X GET "localhost:9200/_cluster/health?pretty"
   ```
   b. You can check node status by running the following command:
   ```
   $ curl -X GET "localhost:9200/_nodes?pretty"
   ```

4.5.2.2 Kibana

4.5.2.2.1 Deploy Kibana

The deployment of Kibana is dependent on the Kubernetes and Elasticsearch clusters to be running. See section 4.6.1.1 for Kibana configuration. Please note, prior to configuring TLS the Kibana password will be the default value of `changeme`. Once Kibana configuration is complete, follow section 4.5.6.1 for deployment.

4.5.2.2.2Configure Index Pattern

After Filebeat is running (4.5.2.4), an Index Pattern will need to be created in order to view data in Kibana.

1. In Kibana, open the **Management tab** and then click **Index Patterns**
2. In the **Index Pattern** text box, enter **filebeat-***
3. If there is already an index created by one of the filebeats, you should see the following text: **Success! Your index pattern matches 1 index.**
4. Click **Next step** and open the **Time Filter field name** dropdown and select **@timestamp**.
5. Click **Create index pattern**.
6. Navigate to the **Discover tab** and verify that logs are there.

4.5.2.3 Transport Layer Security (TLS) and License Upgrade

TLS needs to be configured for Elasticsearch clusters in order to apply a Gold or Platinum license.

4.5.2.3.1 Configure TLS

As the elastic system user (with one exception), perform the following actions on each node (with three exceptions).

1. Create a certificate authority (CA) which is used to sign the certificates of each node:

   ```
   $ elasticsearch-7.4.2/bin/elasticsearch-certutil ca
   ```

   a. This step should take place on only one node.

   b. The command will prompt you for an output file name, leave this blank for the default name.

   c. The command will prompt you for a password, remember this password because it will be used to sign all the nodes' certificates.

   d. The command will output a CA, distribute this file to all nodes so that they can sign their certificate.

2. Generate/sign a certificate:

   ```
   $ elasticsearch-7.4.2/bin/elasticsearch-certutil cert --ca elastic-stack-ca.p12
   ```

   a. `elastic-stack-ca.p12` is the output of the 1st step.

   b. The command will prompt you for a password to the CA, it is the same password set in the 1st step.

   c. The command will prompt you for an output name and password for the signed certificate. Leave these blank.

3. Create a new directory and move the signed certificate there

   ```
   $ mkdir elasticsearch-7.4.2/config/certs
   $ mv elastic-certificates.p12 elasticsearch-7.4.2/config/certs/
   ```

   a. `elastic-certificates.p12` is the output of step 2

4. Remove the CA

   ```
   $ rm elastic-stack-ca.p12
   ```

   a. Perform this step on all nodes except one. If you add new nodes to the cluster in the future, you will need this file.

5. Add the following lines to `elasticsearch-7.4.2/config/elasticsearch.yml`:

   ```
   xpack.security.enabled: true
   xpack.security.transport.ssl.enabled: true
   xpack.security.transport.ssl.verification_mode: certificate
   xpack.security.transport.ssl.keystore.path: certs/elastic-certificates.p12
   xpack.security.transport.ssl.truststore.path: certs/elastic-certificates.p12
   ```

6. As a superuser, restart Elasticsearch:

   ```
   $ sudo systemctl restart elasticsearch.service
   ```

7. Set the password for all built-in users on just one node, save these passwords for future use:

   ```
   $ bin/elasticsearch-setup-passwords interactive
   ```

4.5.2.3.2Upgrade License

Perform the following action on one of the nodes (replace ***<ELASTIC-PASSWORD>*** using the Placeholder Value table and **<LICENSE_FILE>** with the path to the license json file):

```
$ curl -XPUT -u elastic:<ELASTIC-PASSWORD> 'http://localhost:9200/_license' -H
"Content-Type: application/json" -d @<LICENSE_FILE>
```

4.5.2.3.3 Add Elasticsearch Watchers

Elasticsearch Watchers can only be added once the license has been upgraded. Navigate to the machine that contains the source code configured for FDOT. If the Watcher settings have not yet been configured for an environment, see section 4.6.1.3. In the source code directory, navigate to ***Deploy\Monitoring\Watchers*** and perform the following command in a PowerShell terminal:

```
$ Add-Watchers.ps1 -Activate -Environment FDOT
```

Once the script has completed, verify in Kibana that the Watchers were successfully added.

4.5.2.4 Filebeat

4.5.2.4.1Filebeat for Kubernetes Services

The deployment of Filebeat is dependent on the Kubernetes and Elasticsearch clusters to be running. It is also assumed that Kibana and TLS has been configured. See section 4.6.1.1 for Filebeat configuration. Once Kibana configuration is complete, follow section 4.5.6.1 for deployment.

4.5.2.4.2Filebeat for Windows Services

It is also assumed that Kibana and TLS has been configured. Perform the following steps on the Windows authentication server. If any other Windows hosts require filebeat to be deployed, the paths variable might change.

1. Download filebeat:
   ```
   PS> curl https://artifacts.elastic.co/downloads/beats/filebeat/filebeat-7.4.2-windows-x86_64.zip -Outfile filebeat-7.4.2-windows-x86_64.zip
   ```
2. Unzip directory:
   ```
   PS> Expand-Archive filebeat-7.4.2-windows-x86_64.zip -DestinationPath .\
   ```
3. Rename directory:
   ```
   PS> Rename-Item filebeat-7.4.2-windows-x86_64 Filebeat
   ```
4. Move directory to Program Files:
   ```
   PS> Move-Item .\Filebeat ‘C:\Program Files\’
   ```
5. Edit the ***filebeat.yml*** file. Remove the contents and replace it with the following lines:
   ```
   filebeat.inputs:
   - type: container
   ```

```
  enabled: true
  paths:
    - 'C:\ProgramData\docker\containers\*\*.log'
  processors:
  - add_host_metadata: ~
  - add_cloud_metadata: ~
  - decode_json_fields:
      fields: ["message"]
      process_array: false
      max_depth: 1
      target: ""
      overwrite_keys: false

filebeat.config.modules:
  path: ${path.config}/modules.d/*.yml
  reload.enabled: false

setup.template.overwrite: true
setup.template.settings:
  index.number_of_shards: 6
  index.number_of_replicas: 2

output.elasticsearch:
  hosts: ["<HOST-ELASTICSEARCH-0>:9200", "<HOST-ELASTICSEARCH-1>:9200",
"<HOST-ELASTICSEARCH-2>:9200"]
  username: "elastic"
  password: "${ES_PW}"
```

6. Replace the Elasticsearch host placeholders using the Placeholder values table .
7. Create a Windows Service using the provided script in the `Filebeat` directory:
   ```
   PS> .\install-service-filebeat.ps1
   ```
8. Create a filebeat keystore:
   ```
   PS> .\filebeat.exe keystore create
   ```
9. Add a key to the keystore. It will prompt you for a value. Use the Placeholder Values table for the:
   ```
   PS> .\filebeat.exe keystore add ES_PW
   ```
10. Copy the keystore file to `ProgramData\filebeat`:
    ```
    PS> Copy-Item C:\Program Files\Filebeat\data\filebeat.keystore
    C:\ProgramData\filebeat\
    ```
11. Start the filebeat service:
    ```
    PS> Start-Service filebeat
    ```
12. Ensure that the service is running:
    ```
    PS> Get-Service filebeat
    ```

4.5.2.4.3 Filebeat for Mongo Services

On the MongoDB server(s), and any other Linux hosts where metricbeat should be deployed:

1. Download filebeat:

   ```
   $curl -L -O https://artifacts.elastic.co/downloads/beats/filebeat/ filebeat-7.4.2-amd64.deb
   ```

2. Run sudo dpkg to install Filebeat

   ```
   sudo dpkg -i filebeat-7.4.2-amd64.deb
   ```

3. Edit the ***filebeat.yml*** file in /etc/filebeat folder. Remove the contents and replace it with the following lines:

   ```
   filebeat.inputs:
   - type: container
    enabled: true
    paths:
      - /var/log/*.log'
    processors:
    - add_host_metadata: ~
    - add_cloud_metadata: ~
    - decode_json_fields:
        fields: ["message"]
        process_array: false
        max_depth: 1
        target: ""
        overwrite_keys: false

   filebeat.config.modules:
    path: ${path.config}/modules.d/*.yml
    reload.enabled: false

   setup.template.overwrite: true
   setup.template.settings:
    index.number_of_shards: 6
    index.number_of_replicas: 2

   output.elasticsearch:
    hosts: ["<HOST-ELASTICSEARCH-0>:9200", "<HOST-ELASTICSEARCH-1>:9200", "<HOST-ELASTICSEARCH-2>:9200"]
    username: "elastic"
    password: "<ELASTIC-PASSWORD>"
   ```

4. Edit the /etc/filebeat/modules.d/mongodb.yml file. Remove the contents and replace it with the following lines:

```
- module: mongodb
  log:
    enabled: true
    var.paths:
      - /data/mongo/log/mongod.log
```

5. Start the filebeat service:
   ```
   $ service filebeat start
   ```
6. Add filebeat service to systemctl to start automatically on system startup
   ```
   $ systemctl enable filebeat
   ```
7. Ensure that the service is running:
   ```
   $ service filebeat status
   ```

4.5.2.5 Metricbeat

The deployment of Metricbeat is dependent on the Kubernetes and Elasticsearch clusters to be running. The following steps also assume that Kibana and TLS has been configured.

4.5.2.5.1 Metricbeat for Windows Services

On the Windows SQL server, and any other Windows hosts where Metricbeat should be deployed:

1. Download Metricbeat:
   ```
   PS>curl  https://artifacts.elastic.co/downloads/beats/metricbeat/metricbe  at-7.5.1-windows-x86_64.zip -Outfile metricbeat-7.5.1-windows-x86_64.zip
   ```
2. Unzip directory:
   ```
   PS> Expand-Archive metricbeat-7.5.1-windows-x86_64.zip -DestinationPath .\
   ```
3. Rename directory:
   ```
   PS> Rename-Item metricbeat-7.5.1-windows-x86_64 Metricbeat
   ```
4. Move directory to Program Files:
   ```
   PS> Move-Item .\Metricbeat ‘C:\Program Files\’
   ```
5. Edit the ***metricbeat.yml*** file. Remove the contents and replace it with the following lines:
   ```
   metricbeat.config.modules:
     path: ${path.config}/modules.d/*.yml
     reload.enabled: false
   setup.template.settings:
     index.number_of_shards: 6
     index.number_of_replicas: 2
     index.codec: best_compression
   setup.kibana:
   output.elasticsearch:
   ```

```
 hosts: ["<HOST-ELASTICSEARCH-0>:9200", "<HOST-ELASTICSEARCH-1>:9200", "<HOST-ELASTICSEARCH-2>:9200"]
 username: "elastic"
 password: "<ELASTIC-PASSWORD>"
processors:
 - add_host_metadata: ~
 - add_cloud_metadata: ~
 - add_docker_metadata: ~
 - add_kubernetes_metadata: ~
```

6. Edit the modules.d/system-mssql.yml file. Remove the contents and replace it with the following lines:

   ```
   - module: system
    period: 1m
    metricsets:
      - filesystem
    processors:
      - drop_event.when.regexp:
          system.filesystem.device_name: '^[^E]:'
   ```

7. Create a Windows Service using the provided script in the ***Metricbeat*** directory:

   ```
   PS> .\install-service-metricbeat.ps1
   ```

8. Start the metricbeat service:

   ```
   PS> Start-Service metricbeat
   ```

9. Ensure that the service is running:

   ```
   PS> Get-Service metricbeat
   ```

4.5.2.5.2 Metricbeat for Mongo Services

Perform the following steps on the MongoDB server(s):

1. Download Metricbeat:

   ```
   $curl -L -O https://artifacts.elastic.co/downloads/beats/metricbeat/metricbeat-7.5.1-amd64.deb
   ```

2. Run sudo dpkg to install Metricbeat

   ```
   sudo dpkg -i metricbeat-7.5.1-amd64.deb
   ```

3. Edit the ***metricbeat.yml*** file in /etc/metricbeat folder. Remove the contents and replace it with the following lines:

   ```
   metricbeat.config.modules:
    path: ${path.config}/modules.d/*.yml
    reload.enabled: false
   setup.template.settings:
    index.number_of_shards: 6
   ```

```
 index.number_of_replicas: 2
 index.codec: best_compression
setup.kibana:
output.elasticsearch:
 hosts: ["<HOST-ELASTICSEARCH-0>:9200", "<HOST-ELASTICSEARCH-1>:9200", "<HOST-ELASTICSEARCH-2>:9200"]
 username: "elastic"
 password: "<ELASTIC-PASSWORD>"
processors:
 - add_host_metadata: ~
 - add_cloud_metadata: ~
 - add_docker_metadata: ~
 - add_kubernetes_metadata: ~
```

4. Edit the /etc/metricbeat/modules.d/system-mongo.yml file. Remove the contents and replace it with the following lines:

   ```
   - module: system
    period: 1m
    metricsets:
      - filesystem
    processors:
      - drop_event.when.not.equals:
          system.filesystem.mount_point: "/data/mongo"
   ```

5. Start the metricbeat service:

   ```
   $ service metricbeat start
   ```

6. Add metricbeat service to systemctl to start automatically on system startup

   ```
   $ systemctl enable metricbeat
   ```

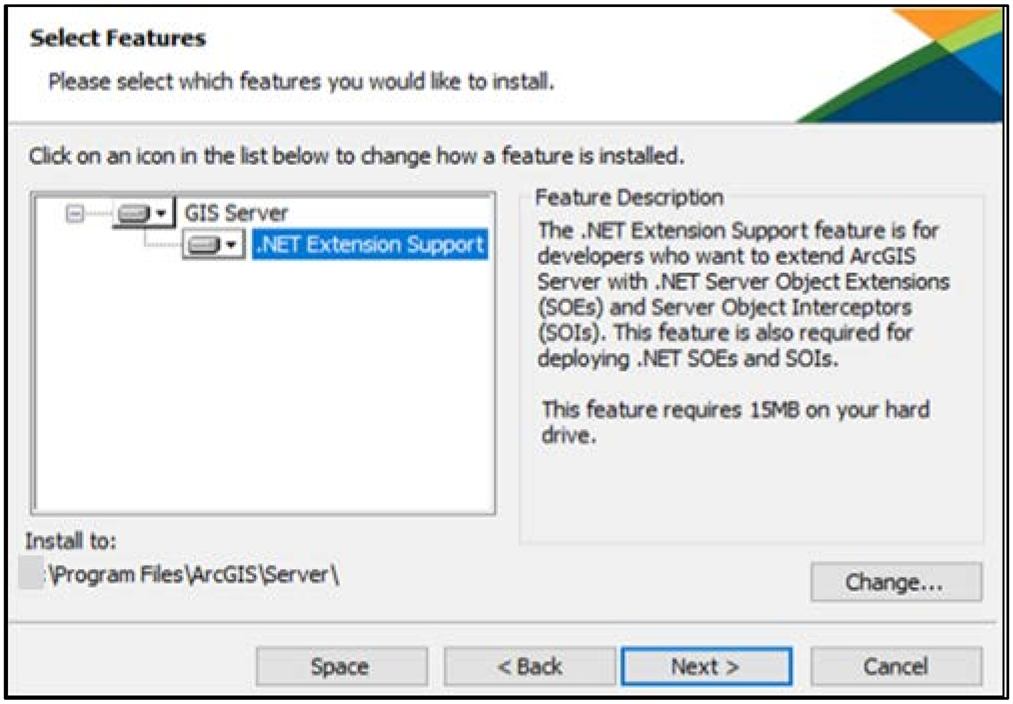
7. Ensure that the service is running:

   ```
   $ service metricbeat status
   ```

4.5.3 GIS Installation

4.5.3.1 GeoEvent Server Installation:

1. Unpack the ArcGIS Server 10.7.1 software in the package downloaded from https://my.esri.com to <HOST_GEOEVENTSERVER>
2. Install the package to the desired drive



Select Features

Please select which features you would like to install.

Click on an icon in the list below to change how a feature is installed.

- GIS Server
  - .NET Extension Support

Feature Description

The .NET Extension Support feature is for developers who want to extend ArcGIS Server with .NET Server Object Extensions (SOEs) and Server Object Interceptors (SOIs). This feature is also required for deploying .NET SOEs and SOIs.

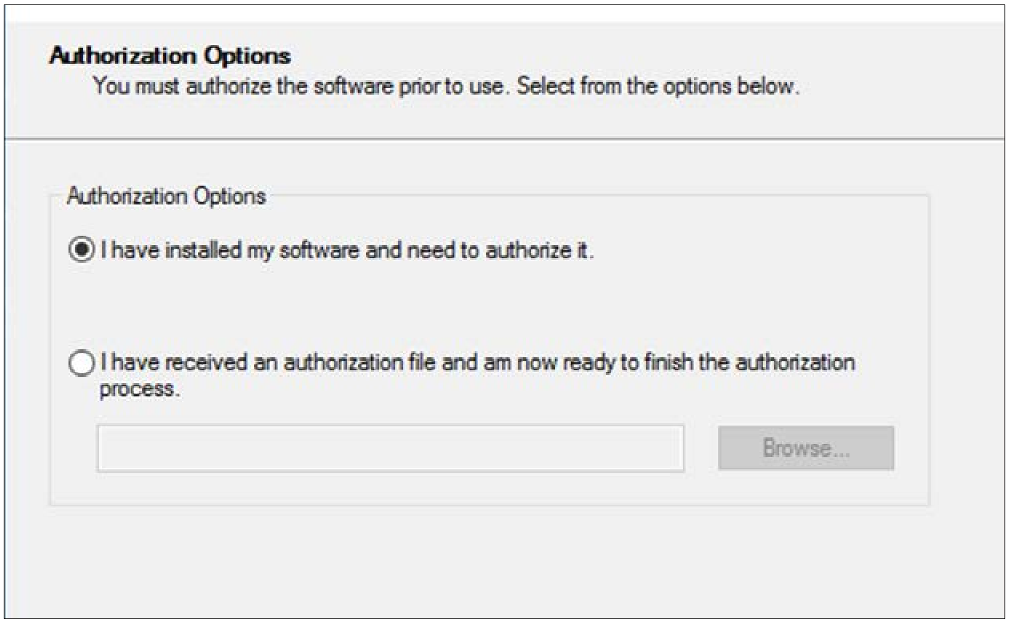
This feature requires 15MB on your hard drive.

Install to:

\Program Files\ArcGIS\Server\

Figure 2 – ArcGIS Server Installation Setup

3. Provide the RICMS Operations and Maintenance account credentials <USERNAME-AD-SERVICE-ACCOUNT>/<PASSWORD-AD-SERVICE-ACCOUNT>
4. Complete installation of ArcGIS Server by authorizing using the GeoEvent Server license file.



Authorization Options

You must authorize the software prior to use. Select from the options below.

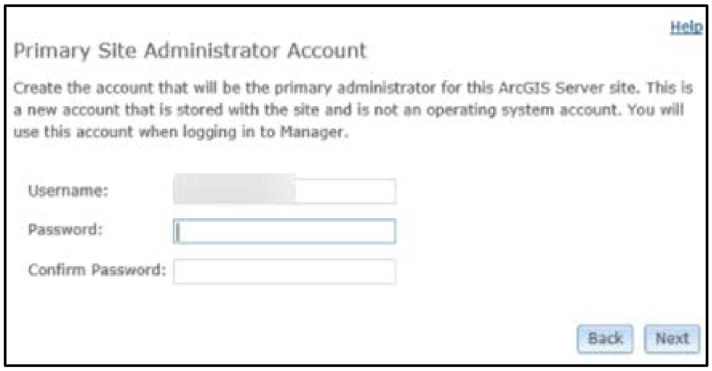
Authorization Options

● I have installed my software and need to authorize it.

○ I have received an authorization file and am now ready to finish the authorization process.

Figure 3 – ArcGIS Server Authorization

5. Navigate to Start→ArcGIS→ArcGIS Server Manager URL. This will redirect the user to the “Create new site” page of the Server Manager.



Primary Site Administrator Account

Create the account that will be the primary administrator for this ArcGIS Server site. This is a new account that is stored with the site and is not an operating system account. You will use this account when logging in to Manager.

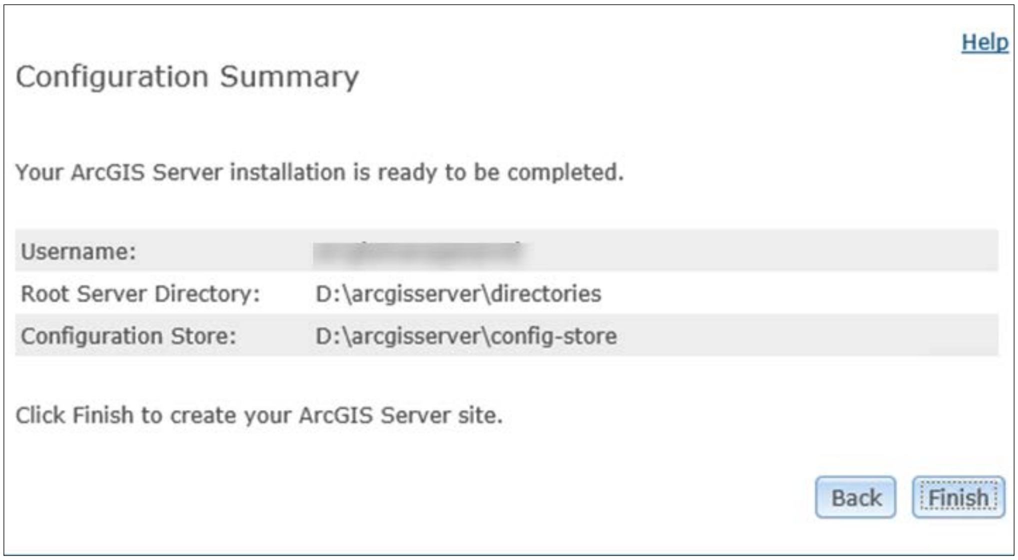
Username: [redacted]

Password:

Confirm Password:

Figure 4 – ArcGIS Server Site Account Creation

6. Create a new ArcGIS Server site, and administrator account for server manager using the RICMS Operations and Maintenance account credentials <USERNAME-AD-SERVICE-ACCOUNT>/<PASSWORD-AD-SERVICE-ACCOUNT>



Configuration Summary

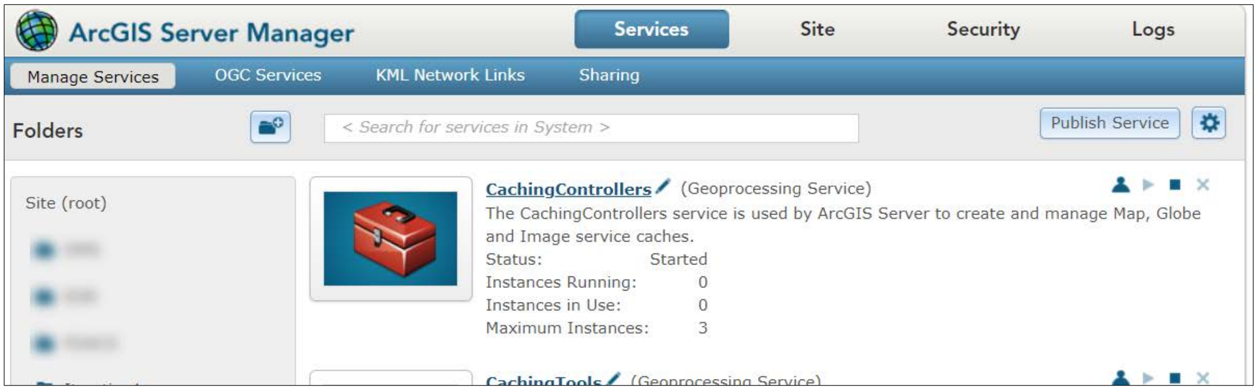
Your ArcGIS Server installation is ready to be completed.

Username:	[redacted]
Root Server Directory:	D:\arcgisserver\directories
Configuration Store:	D:\arcgisserver\config-store

Click Finish to create your ArcGIS Server site.

Figure 5 – ArcGIS Server Site Creation Summary

7. Once the site is created, use the <USERNAME-AD-SERVICE-ACCOUNT>/<PASSWORD-AD-SERVICE-ACCOUNT>to login to the server manager.



CachingControllers (Geoprocessing Service)

The CachingControllers service is used by ArcGIS Server to create and manage Map, Globe and Image service caches.

Status: Started

Instances Running: 0

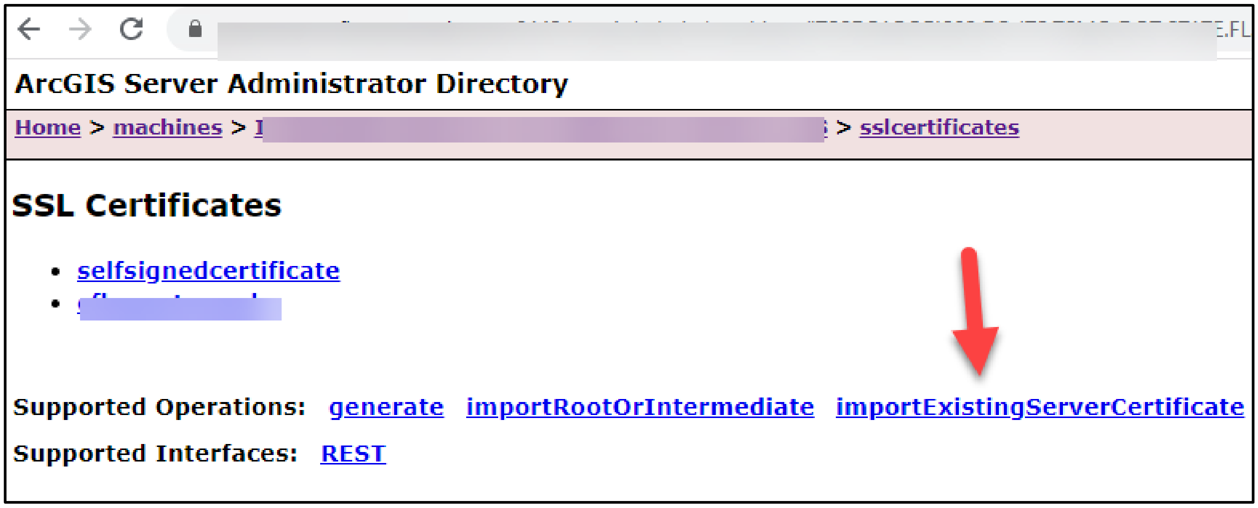
Instances in Use: 0

Maximum Instances: 3

CachingTools (Geoprocessing Service)

Figure 6 – ArcGIS Server Manager

8. Follow the steps in the section 4.5.3.2 to install ‘Web Adaptor for Server’ and obtain the URL for Server Manager on GeoEvent machine as: <GEOEVENT-SERVER-MANAGER-URL>.
9. Unpack the ArcGIS GeoEvent Server 10.7.1 software in the package downloaded from https://my.esri.com to <HOST_GEOEVENTSERVER>
10. Install the package to the desired drive
11. Complete installation of ArcGIS Server by authorizing using the GeoEvent Server license file.
12. Navigate to Start→ArcGIS→GeoEvent Manager URL. This will redirect the user to the <GEOEVENTMANAGER-URL>
13. Import appropriate server certificate to the <GEOEVENT-SERVER-MANAGER-URL>



[redacted]

ArcGIS Server Administrator Directory

Home > machines > [redacted] > sslcertificates

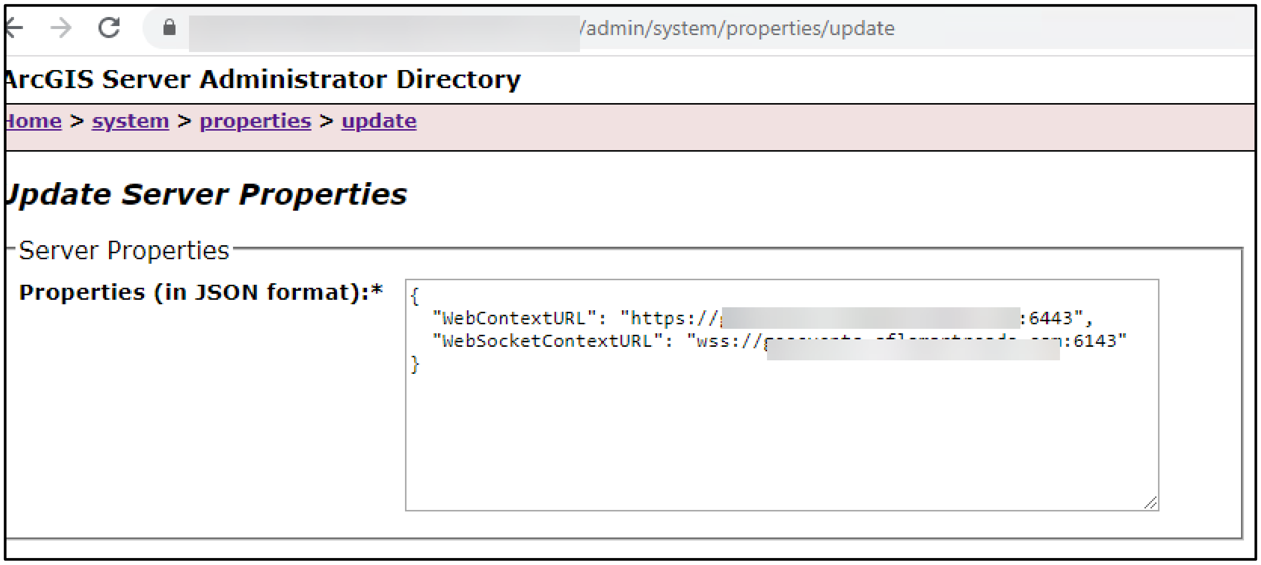
SSL Certificates

- selfsignedcertificate
- [redacted]

Supported Operations: generate importRootOrIntermediate importExistingServerCertificate

Supported Interfaces: REST

14. Update the ‘WebContextURL’ and ‘WebSocketContextURL’ properties in the Server Manager admin to <WEBCONTEXT-PROPERTIES>



[redacted]/admin/system/properties/update

ArcGIS Server Administrator Directory

Home > system > properties > update

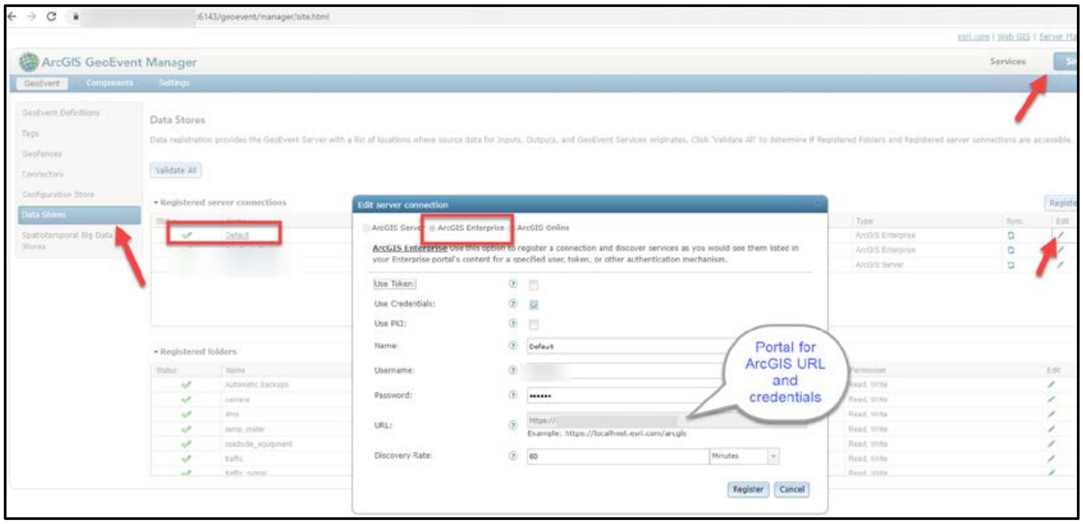
Update Server Properties

Server Properties

Properties (in JSON format):*

```
{
  "WebContextURL": "https://[redacted]:6443",
  "WebSocketContextURL": "wss://[redacted]:6143"
}
```

15. Federate ArcGIS Server Manager on the GeoEvent machine with Portal.
16. Update ‘Default’ Data Store to ArcGIS Enterprise on the GeoEvent Manager using Portal credentials



Edit server connection

○ ArcGIS Server ● ArcGIS Enterprise ○ ArcGIS Online

ArcGIS Enterprise Use this option to register a connection and discover services as you would see them listed in your Enterprise portal's content for a specified user, token, or other authentication mechanism.

Use Token: □

Use Credentials: ☑

Use PKI: □

Name: Default

Username:

Password: ••••••

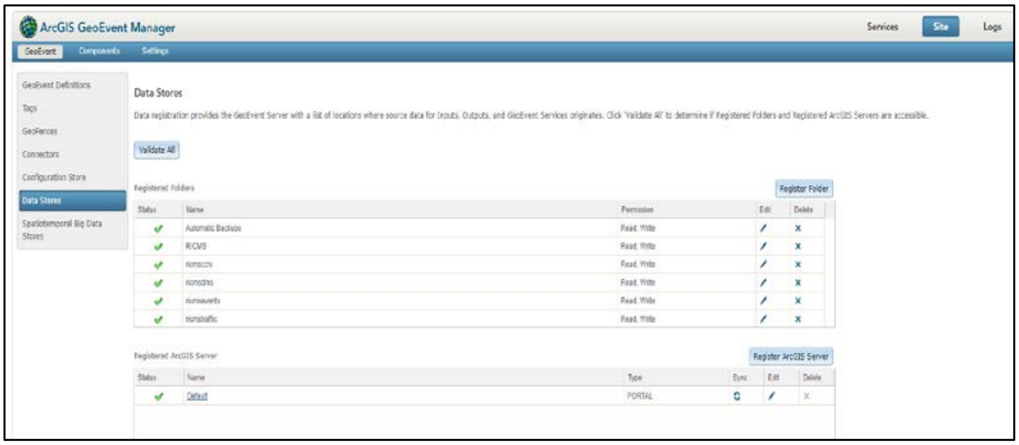
URL: https://

Example: https://localhost.esri.com/arcgis

Discovery Rate: 60 Minutes

Register Cancel

17. Once the GeoEvent Server is federated, use the Portal for ArcGIS credentials to login to GeoEvent Manager. Navigate and verify access to <PORTAL-GIS>, <SUNSTORE-GIS-SERVERMANAGER> URLs from <HOST_ARCGISDESKTOP>, <HOST_GEOEVENTSERVER> machines using <USERNAME-AD-SERVICE-ACCOUNT>, and <PASSWORD-AD-SERVICE-ACCOUNT>.



Data Stores

Data registration provides the GeoEvent Server with a list of locations where source data for Inputs, Outputs, and GeoEvent Services originates. Click 'Validate All' to determine if Registered Folders and Registered ArcGIS Servers are accessible.

Registered Folders

Status	Name	Permission	Edit	Delete
✓	Automatic Backups	Read, Write	✎	x
✓	RICMS	Read, Write	✎	x
✓	ricmsccv	Read, Write	✎	x
✓	ricmsdms	Read, Write	✎	x
✓	ricmsevents	Read, Write	✎	x
✓	ricmstraffic	Read, Write	✎	x

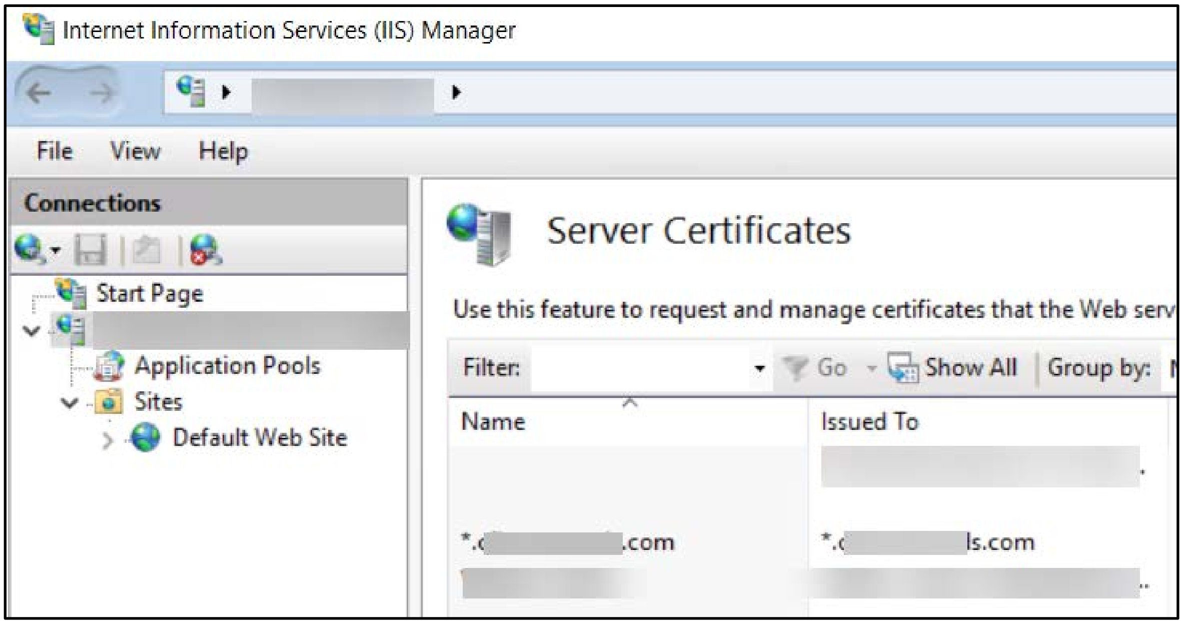
Registered ArcGIS Server

Status	Name	Type	Sync	Edit	Delete
✓	Default	PORTAL	↻	✎	x

Figure 7 – ArcGIS GeoEvent Manager

4.5.3.2 Installation Steps for Web Adaptor for the Server

1. Verify IIS is up and running on <HOST_GEOEVENTSERVER>.
2. Obtain the <SSL-CERITIFICATE-WITH-FQDN> details and import this into IIS.
3. Verify the <SSL-CERITIFICATE-WITH-FQDN> is imported by navigating to the machine name on the left-side of tree in IIS→ Server Certificates (under the 'IIS' section).

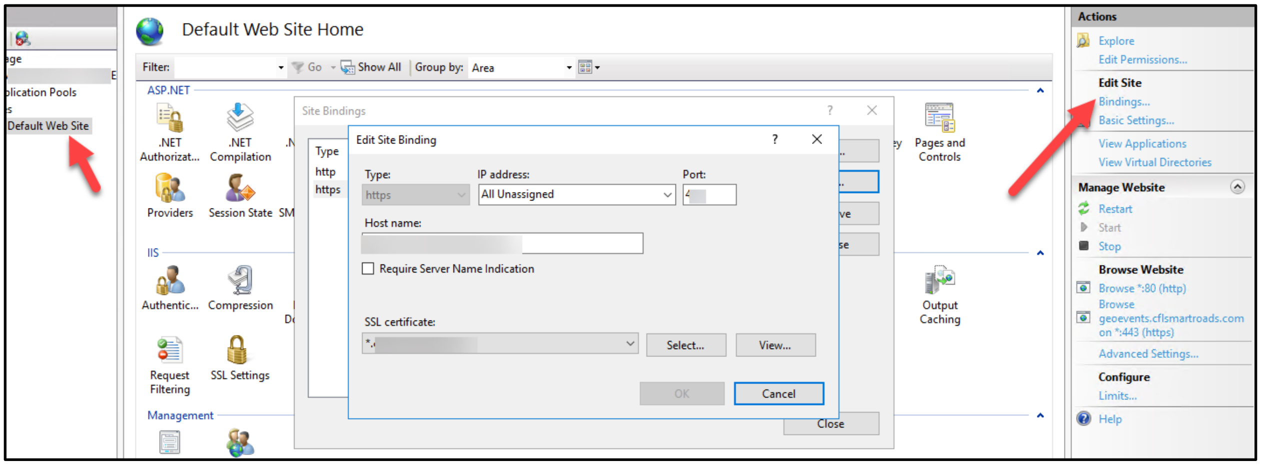


Use this feature to request and manage certificates that the Web serv

Name	Issued To
*.c .com	*.c ls.com

Figure 8 – Verify Certificate from IIS

4. Update the ‘Bindings’ to <GEOEVENTSERVER-DNS>



Edit Site Binding

Type: https

IP address: All Unassigned

Port: 4

Host name:

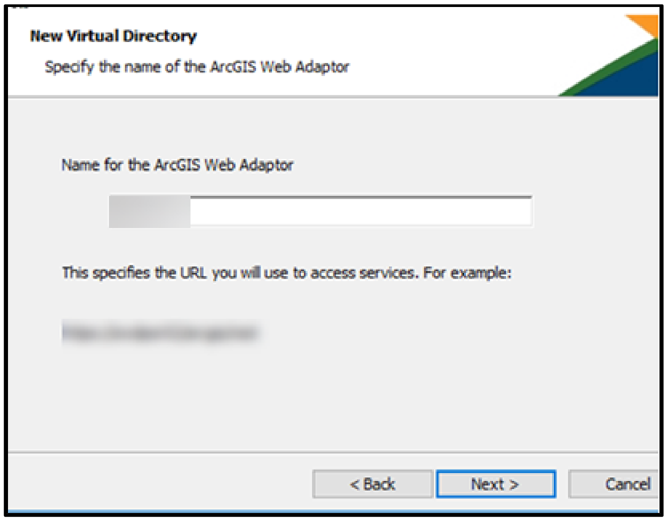
☐ Require Server Name Indication

SSL certificate: *.

Select... View...

OK Cancel

5. Run the ArcGIS Web Adaptor setup.exe file downloaded from http://my.esri.com
6. Provide the name for adaptor <GEOEVENT-WEB-ADAPTOR-NAME> and follow the instructions on the screen.



New Virtual Directory

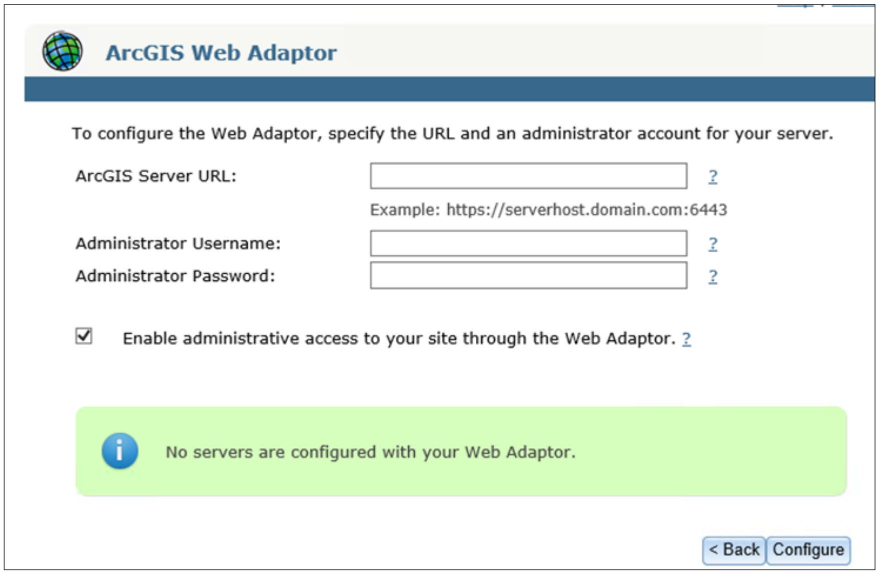
Specify the name of the ArcGIS Web Adaptor

Name for the ArcGIS Web Adaptor

This specifies the URL you will use to access services. For example:

Figure 9 – Web Adaptor for Server Setup

7. Provide <GEOEVENT-SERVER-MANAGER-URL> and provide the credentials <USERNAME-AD-SERVICE-ACCOUNT> and <PASSWORD-AD-SERVICE-ACCOUNT> in the respective textboxes



ArcGIS Web Adaptor

To configure the Web Adaptor, specify the URL and an administrator account for your server.

ArcGIS Server URL: ?

Example: https://serverhost.domain.com:6443

Administrator Username: ?

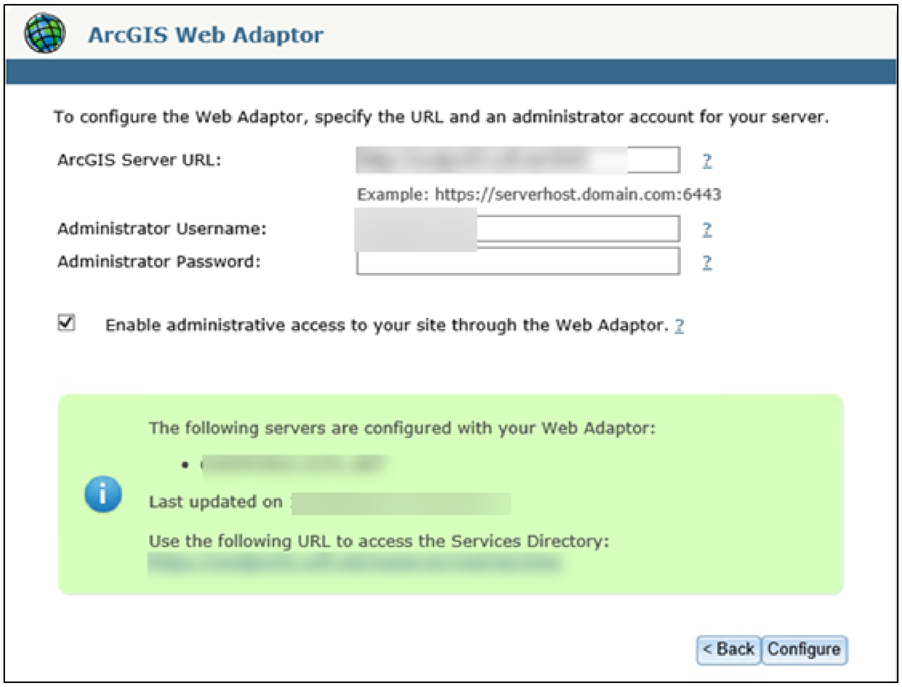
Administrator Password: ?

☑ Enable administrative access to your site through the Web Adaptor. ?

No servers are configured with your Web Adaptor.

Figure 10 – Web Adaptor for Server Configuration

8. Click the configure button, it should display the following message:



To configure the Web Adaptor, specify the URL and an administrator account for your server.

ArcGIS Server URL: [redacted] ?

Example: https://serverhost.domain.com:6443

Administrator Username: [redacted] ?

Administrator Password: ?

☑ Enable administrative access to your site through the Web Adaptor. ?

The following servers are configured with your Web Adaptor:

- [redacted]

Last updated on [redacted]

Use the following URL to access the Services Directory:

[redacted]

Figure 11 – Web Adaptor for Server Configuration Success Message

9. To validate, access the server manager URL with the credentials <USERNAME-AD-SERVICE-ACCOUNT> and <PASSWORD-AD-SERVICE-ACCOUNT>.

4.5.3.3 ArcGIS Desktop Installation

1. Download the ArcGIS Desktop 10.7.1 exe to <HOST_ARCGISDESKTOP> from https://my.esri.com
2. Install and run the setup as Administrator on the machine.
3. Configure the license after successful installation.
4. Navigate to Start→ArcGIS→ArcMap and verify the setup.

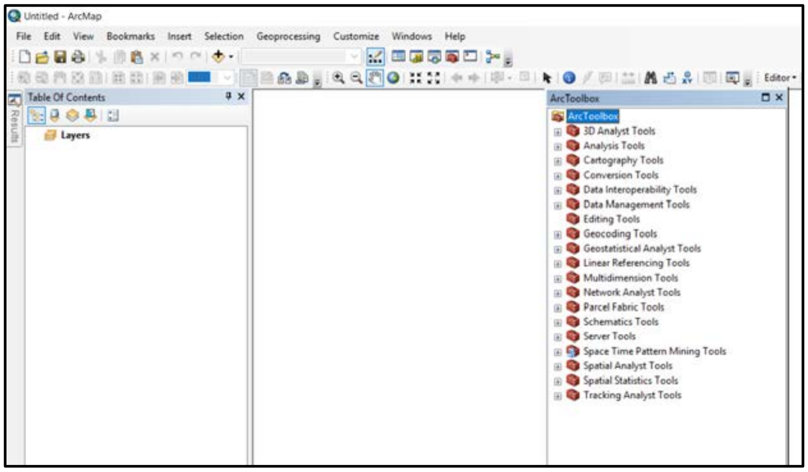


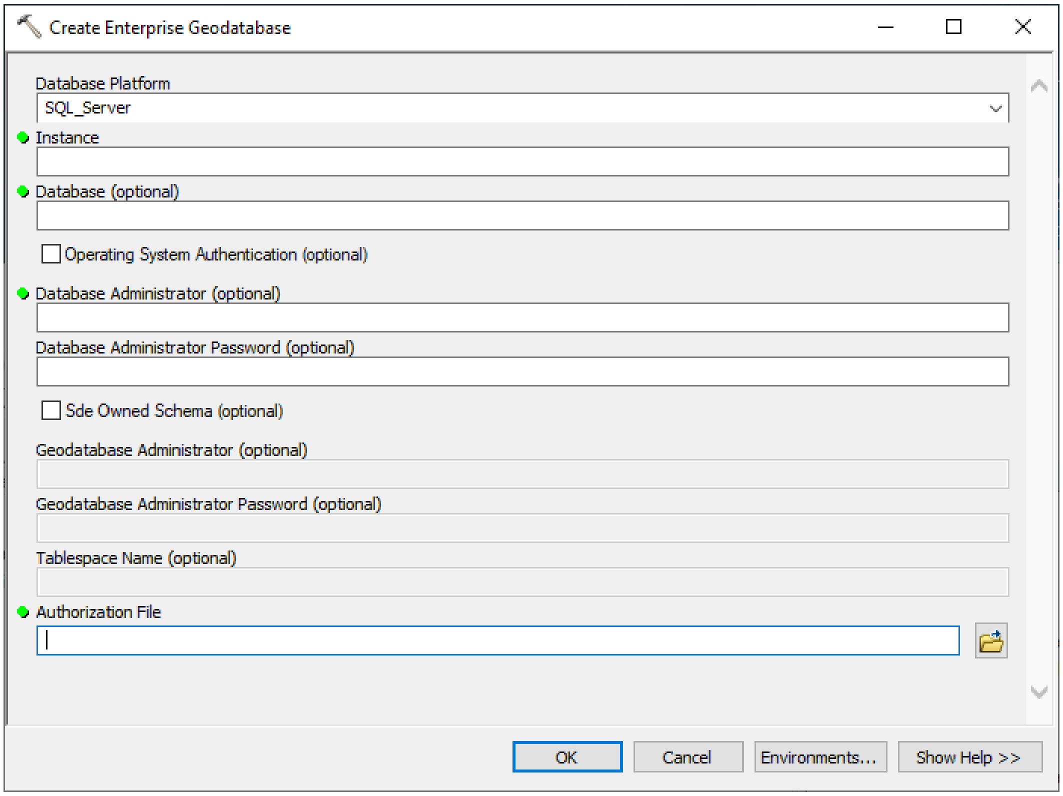
Figure 12 – ArcMap

4.5.3.4 Project Directory Setup

1. Copy all contents from Source Code Tools\gis to <HOST_ARCSERVER> C:\projects\ricms

4.5.3.5 SDE Configuration

1. Navigate to Start > ArcGIS > ArcCatalog
2. Open ArcToolbox
3. Expand: Data Management Tools > Geodatabase Administration
4. Select Create Enterprise Geodatabase
5. Enter SQL Server Instance Info:
    a. Select SQL_Server
    b. Instance: <SDE-INSTANCE>
    c. Database: <SDE-DB>
    d. Uncheck Operating System Authentication
    e. Database Administrator: <SDE-USERNAME>
    f. Database Administrator Password: <SDE-PASSWORD>
    g. Uncheck Sde Owned Schema
    h. Authorization File: Navigate/Point to an active ArcServer Key Codes File (*.ecp)



Create Enterprise Geodatabase

Database Platform: SQL_Server

Instance:

Database (optional):

☐ Operating System Authentication (optional)

Database Administrator (optional):

Database Administrator Password (optional):

☐ Sde Owned Schema (optional)

Geodatabase Administrator (optional):

Geodatabase Administrator Password (optional):

Tablespace Name (optional):

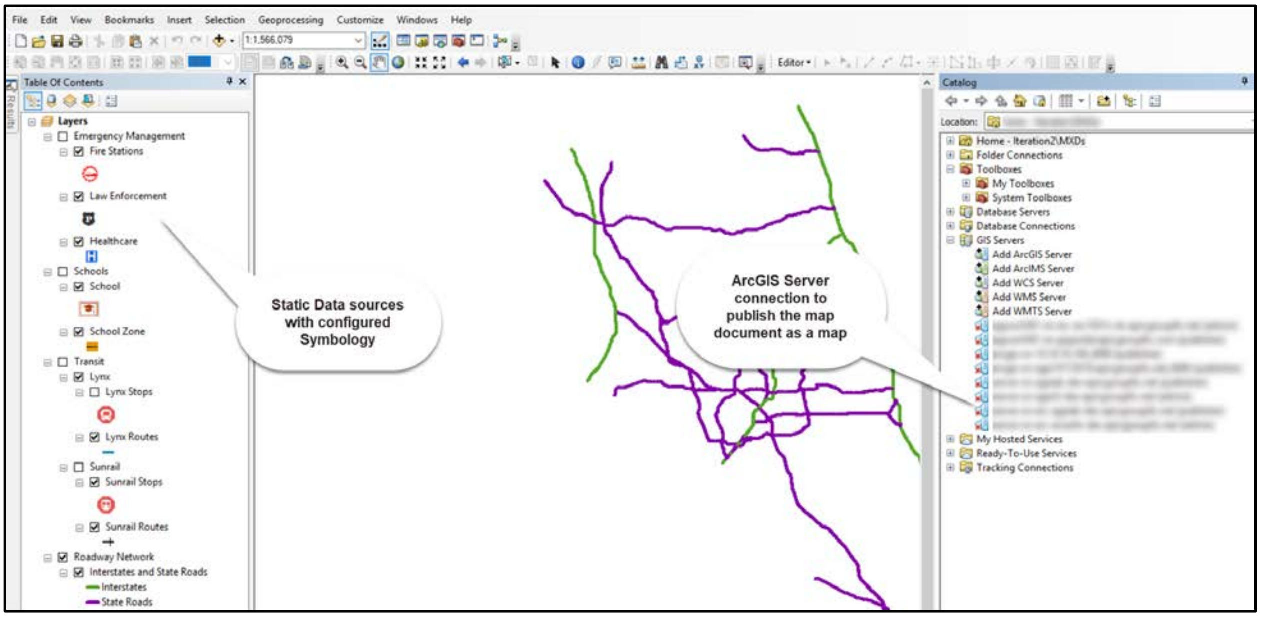
Authorization File:

6. Verify Completion/Success dialog appears when completed

7. Use ArcCatalog > Create Database Connection to SDE
8. Import SDE Workspace Document from c:\projects\ricms\sde\RicmsSde.xml

4.5.3.6 GIS Map/Feature Services Configuration

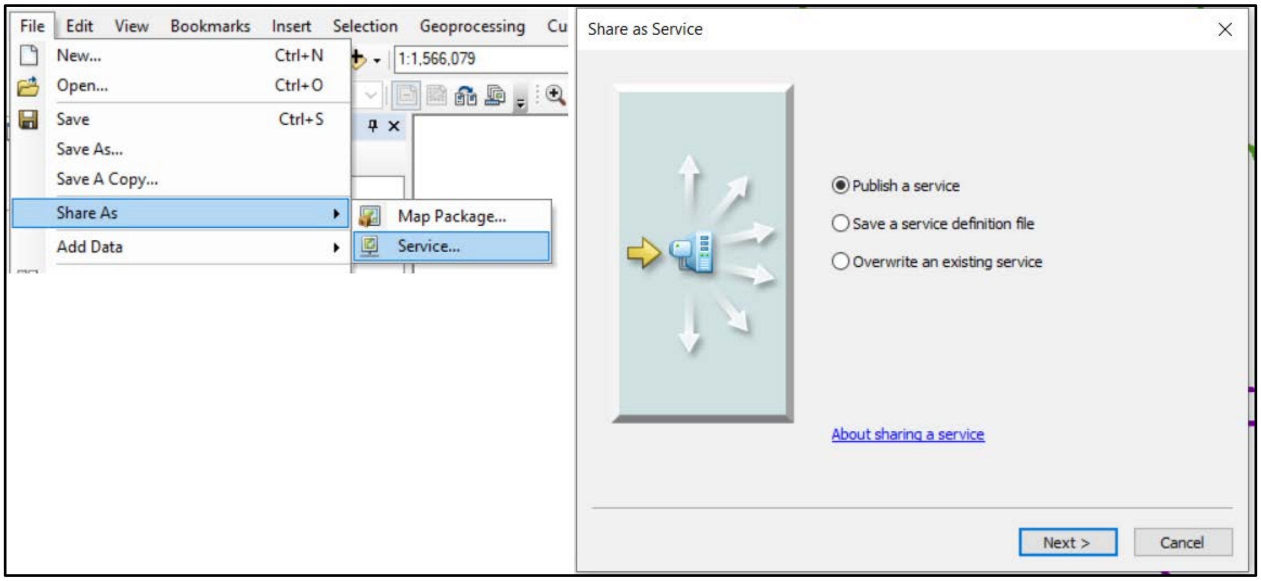
1. Navigate to Start→ArcMap
2. Open the static and event MXD from c:\projects\ricms\projects
   a. Reconnect each layer data source to the appropriate data object in the SDE
   b. Publish each MXD to the ArcServer as Map Services



Static Data sources with configured Symbology

ArcGIS Server connection to publish the map document as a map

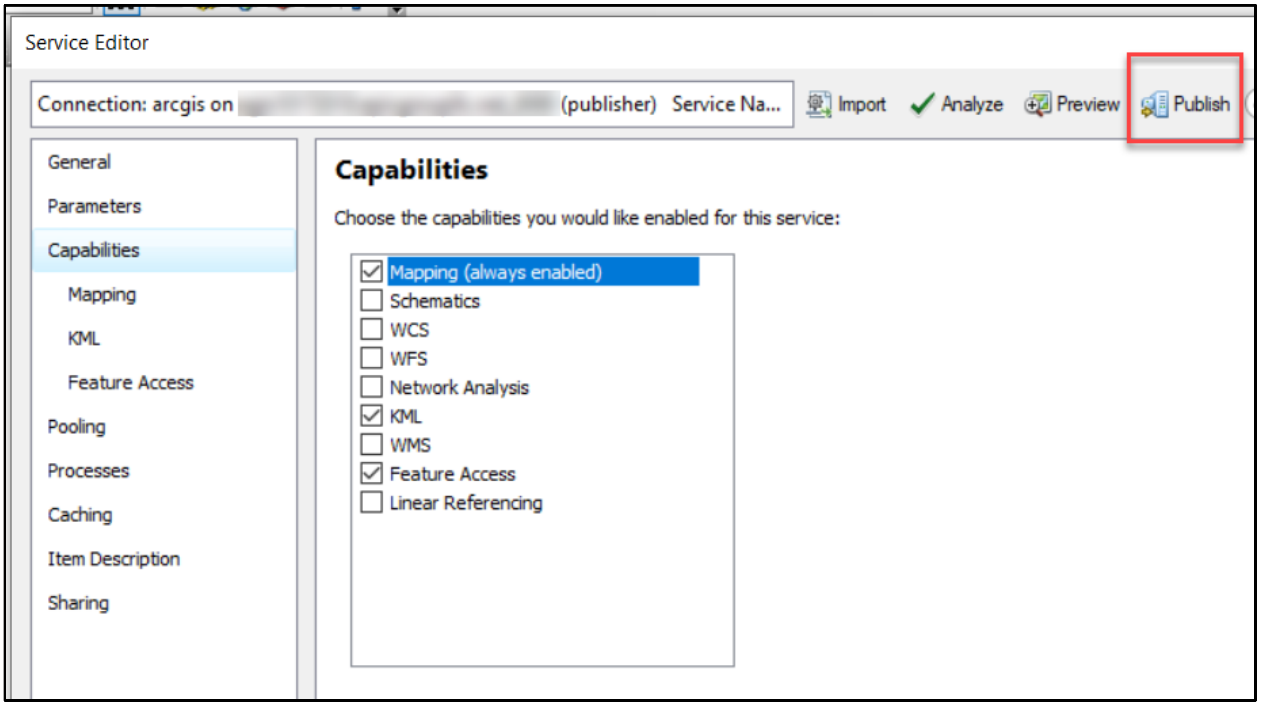
Figure 13 – Static Data Sources Map Document



● Publish a service
○ Save a service definition file
○ Overwrite an existing service

Figure 14 – Publish Map Service

3. Open the ricms, event, and sotMXDs
    a. Reconnect each layer data source to the appropriate data object in the SDE
4. Publish as respective Feature Services by checking the ‘Feature Access’ checkbox.



Service Editor

Connection: arcgis on [redacted] (publisher) Service Na...

Capabilities

Choose the capabilities you would like enabled for this service:

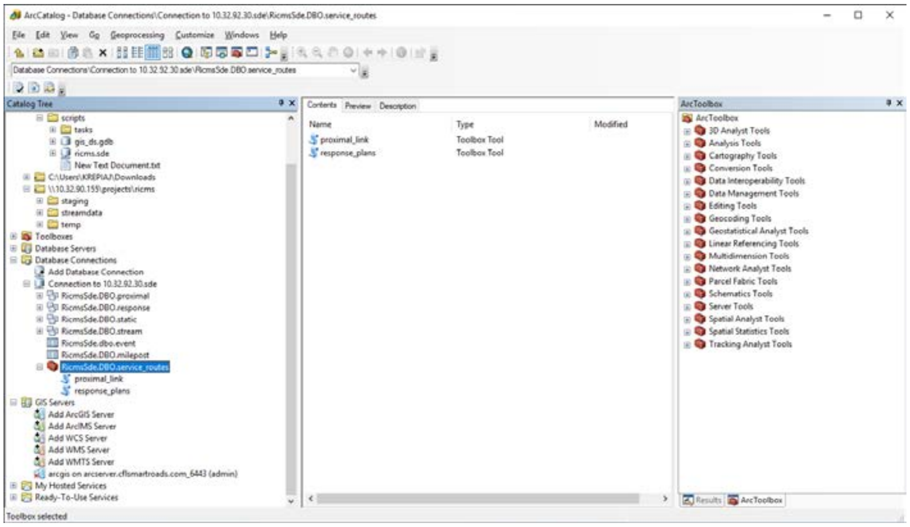
☑ Mapping (always enabled)
☐ Schematics
☐ WCS
☐ WFS
☐ Network Analysis
☑ KML
☐ WMS
☑ Feature Access
☐ Linear Referencing

Figure 15 – Enable 'Feature Access' for Feature Service

4.5.3.7 GIS Support Services Configuration

4.5.3.7.1 Proximal Link

1. Open ArcCatalog > Access the RicmsSde Database Connection
2. Open the service_route Toolbox



ArcCatalog - Database Connections\Connection to 10.32.92.30.sde\RicmsSde.DBO.service_routes

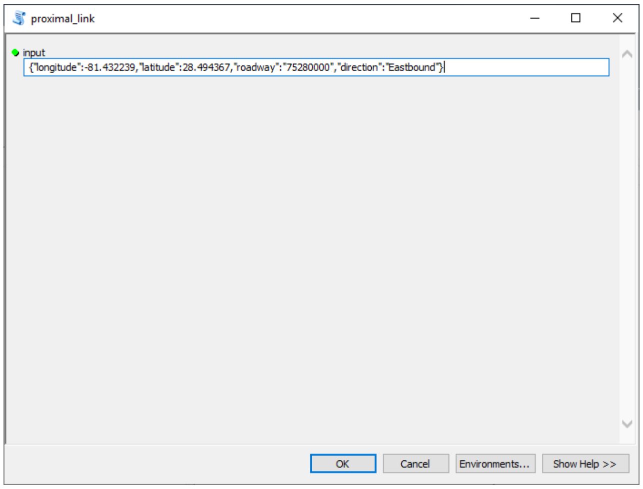
Database Connections\Connection to 10.32.92.30.sde\RicmsSde.DBO.service_routes

Name	Type	Modified
proximal_link	Toolbox Tool	
response_plans	Toolbox Tool	

Toolbox selected

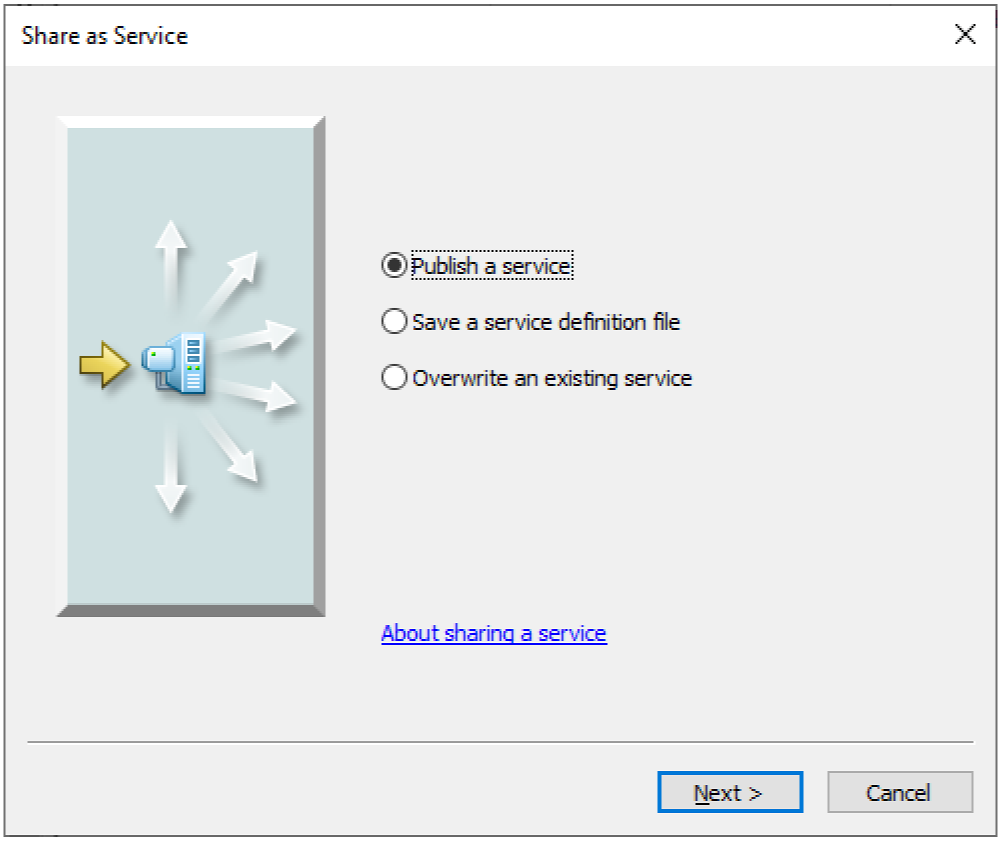
3. Double-click proximal_link Toolbox Tool
4. Paste the following into the input box:

```
{"longitude":-81.432239,"latitude":28.494367,"roadway":"75280000","direction":"Eastbound"}
```



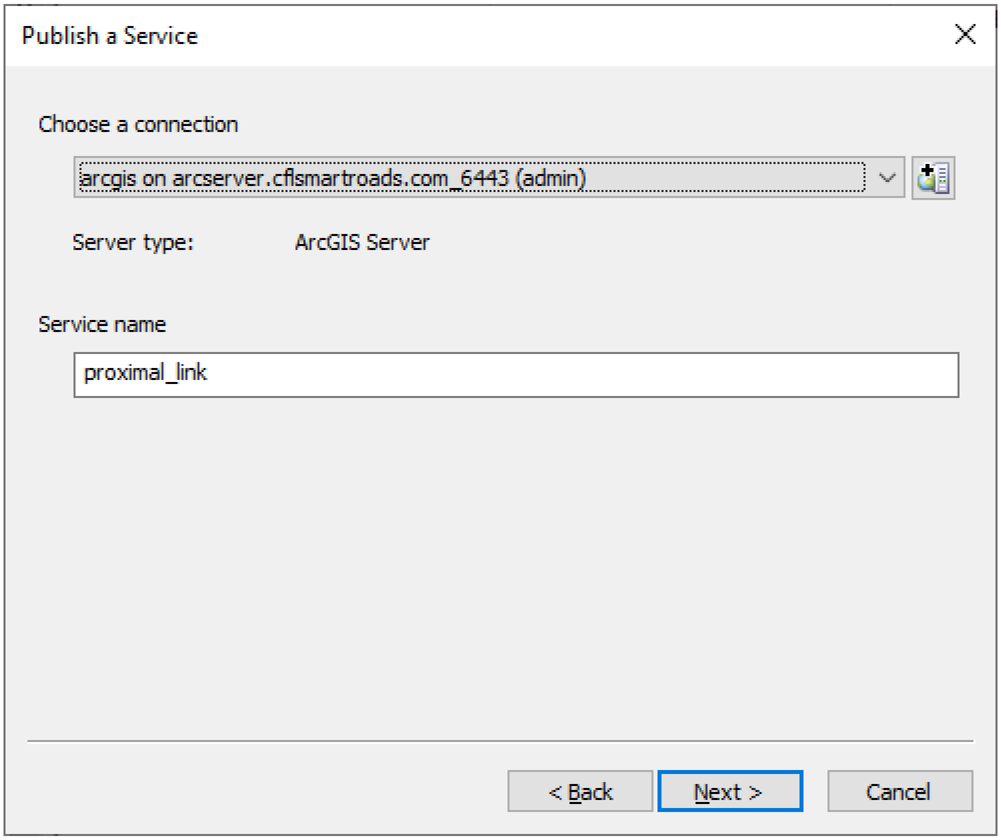
input
{"longitude":-81.432239,"latitude":28.494367,"roadway":"75280000","direction":"Eastbound"}

5. Click OK
6. Access the Results Window > Expand the Current Session Tree Node
7. Right-click the proximal_link results > Share As > Geoprocessing Service
8. Select Publish a service > Click Next



● Publish a service
○ Save a service definition file
○ Overwrite an existing service

9. Select the ArcServer Connection > Set Service Name to “proximal_link” > Click Next



Publish a Service

Choose a connection

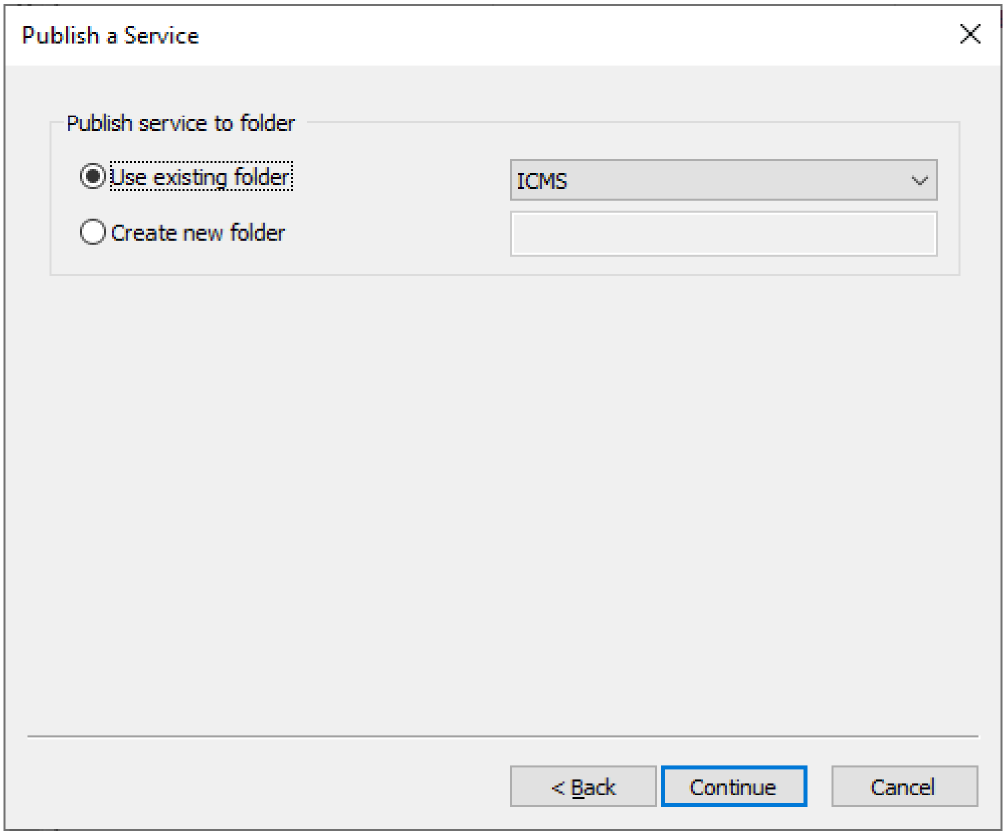
arcgis on arcserver.cflsmartroads.com_6443 (admin)

Server type: ArcGIS Server

Service name

proximal_link

10. Select Use existing folder > Select ICMS > Click Continue



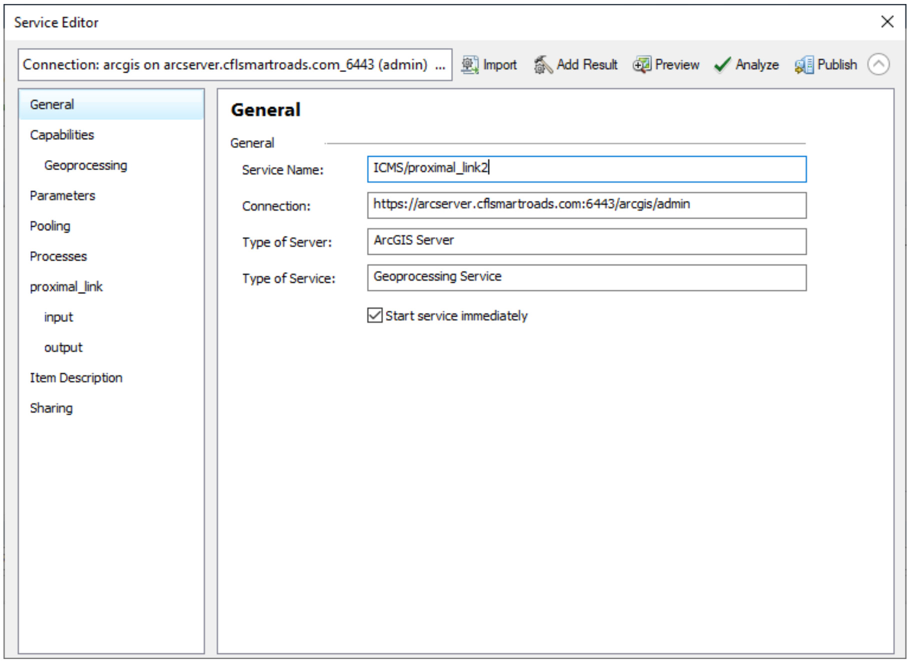
Publish a Service

Publish service to folder

● Use existing folder ICMS

○ Create new folder

11. Click Publish



Connection: arcgis on arcserver.cflsmartroads.com_6443 (admin) ...

Service Name: ICMS/proximal_link2

Connection: https://arcserver.cflsmartroads.com:6443/arcgis/admin

Type of Server: ArcGIS Server

Type of Service: Geoprocessing Service

☑ Start service immediately

4.5.3.7.2Response Plans

1. Open ArcCatalog > Access the RicmsSde Database Connection
2. Open the service_route Toolbox
3. Double-click response_plans Toolbox Tool
4. Paste the following into the input box:
   16997073,17009186
5. Click OK
6. Access the Results Window > Expand the Current Session Tree Node
7. Right-click the response_plans results > Share As > Geoprocessing Service
8. Select Publish a service > Click Next
9. Select the ArcServer Connection > Set Service Name to “response_plans” > Click Next
10. Select Use existing folder > Select ICMS > Click Continue
11. Click Publish

4.5.3.7.3SOT Corridor Plans

1. Open ArcCatalog > Access the RicmsSde Database Connection
2. Open the sot_corridor Toolbox
3. Double-click sot_corridor Toolbox Tool
4. No Default Input needed
5. Click OK
6. Access the Results Window > Expand the Current Session Tree Node

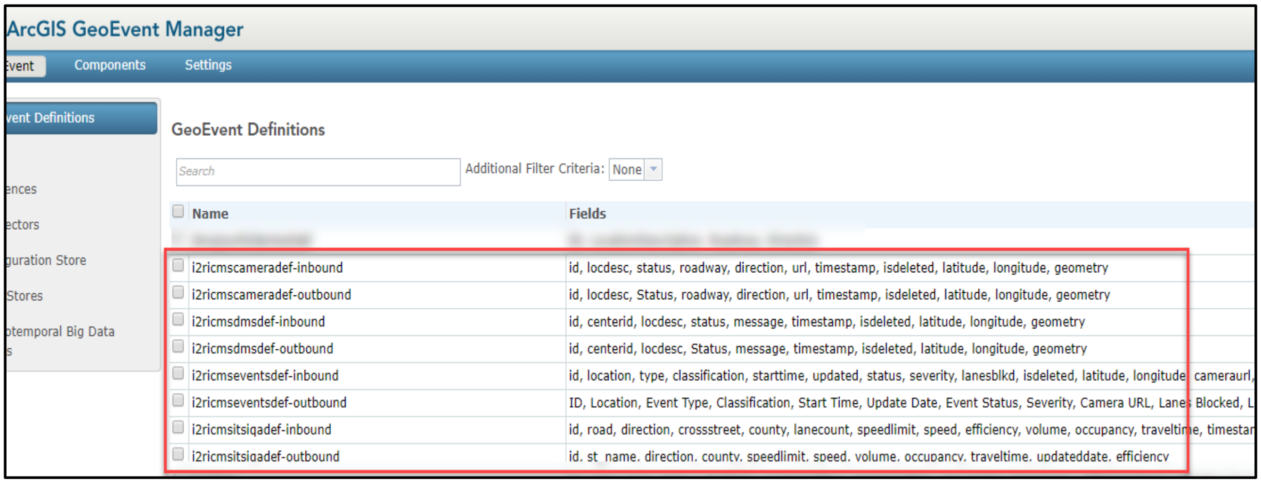
7. Right-click the sot_corridor results > Share As > Geoprocessing Service
8. Select Publish a service > Click Next
9. Select the ArcServer Connection > Set Service Name to “sot_corridor” > Click Next
10. Select Use existing folder > Select ICMS > Click Continue
11. Click Publish

4.5.3.8 XML File System Processing

1. Navigate to <HOST_ARCSERVER> c:\projects\ricms
2. Copy the following folders to <HOST_GEOEVENTSERVER> C:\projects\ricms:
    a. staging
    b. streamdata
3. Access Task Scheduler
4. Click Action > Import Task
5. Navigate to <HOST_ARCSERVER> c:\projects\ricms\tasks
6. Import/Repeat for each of the following tasks:
    a. process-camera.xml
    b. process-dms.xml
    c. process-rampmeter.xml
    d. process-rse.xml
    e. process-tam.xml
    f. process-trafficsignal.xml
    g. process-transit.xml
    h. process-truckparking.xml
    i. process-vehicleposition.xml

4.5.3.9 GeoEvent Services Setup

1. Open the <GEOEVENTMANAGER-URL> and navigate to Site→GeoEvent Definitions to set up the GeoEvent definitions for DMS, Cameras, Events, Traffic Conditions



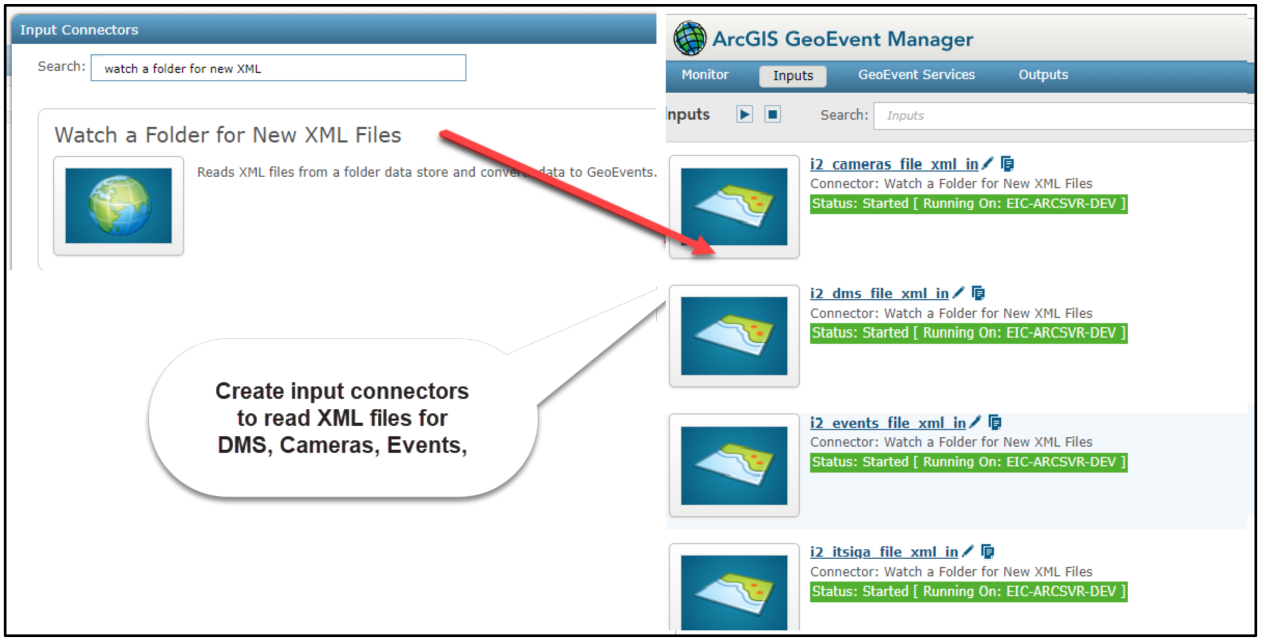
GeoEvent Definitions

Search | Additional Filter Criteria: None

□	Name	Fields
□	[redacted]	[redacted]
□	i2ricmscameradef-inbound	id, locdesc, status, roadway, direction, url, timestamp, isdeleted, latitude, longitude, geometry
□	i2ricmscameradef-outbound	id, locdesc, Status, roadway, direction, url, timestamp, isdeleted, latitude, longitude, geometry
□	i2ricmsdmsdef-inbound	id, centerid, locdesc, status, message, timestamp, isdeleted, latitude, longitude, geometry
□	i2ricmsdmsdef-outbound	id, centerid, locdesc, Status, message, timestamp, isdeleted, latitude, longitude, geometry
□	i2ricmseventsdef-inbound	id, location, type, classification, starttime, updated, status, severity, lanesblkd, isdeleted, latitude, longitude, cameraurl,
□	i2ricmseventsdef-outbound	ID, Location, Event Type, Classification, Start Time, Update Date, Event Status, Severity, Camera URL, Lanes Blocked, L
□	i2ricmsitsiqadef-inbound	id, road, direction, crossstreet, county, lanecount, speedlimit, speed, efficiency, volume, occupancy, traveltime, timestam
□	i2ricmsitsiqadef-outbound	id, st_name, direction, county, speedlimit, speed, volume, occupancy, traveltime, updateddate, efficiency

Figure 16- GeoEvent Definitions for Dynamic Data Sources

2. Configure the ‘Watch a Folder for New XML files’ input connector for the dynamic data sources by connecting to the folders created in section 4.5.7 for the respective data sources.



Search: watch a folder for new XML

Watch a Folder for New XML Files

Reads XML files from a folder data store and converts data to GeoEvents.

i2_cameras_file_xml_in
Connector: Watch a Folder for New XML Files
Status: Started [Running On: EIC-ARCSVR-DEV]

i2_dms_file_xml_in
Connector: Watch a Folder for New XML Files
Status: Started [Running On: EIC-ARCSVR-DEV]

i2_events_file_xml_in
Connector: Watch a Folder for New XML Files
Status: Started [Running On: EIC-ARCSVR-DEV]

i2_itsiqa_file_xml_in
Connector: Watch a Folder for New XML Files
Status: Started [Running On: EIC-ARCSVR-DEV]

Create input connectors to read XML files for DMS, Cameras, Events,

Figure 17 - Input connectors to read dynamic data sources XML files

3. Configure Send Features to a Stream Service output connector for DMS, Cameras, Ramp Meters, Roadside Equipment, Traffic Signals, Truck Parking, Vehicle Position.

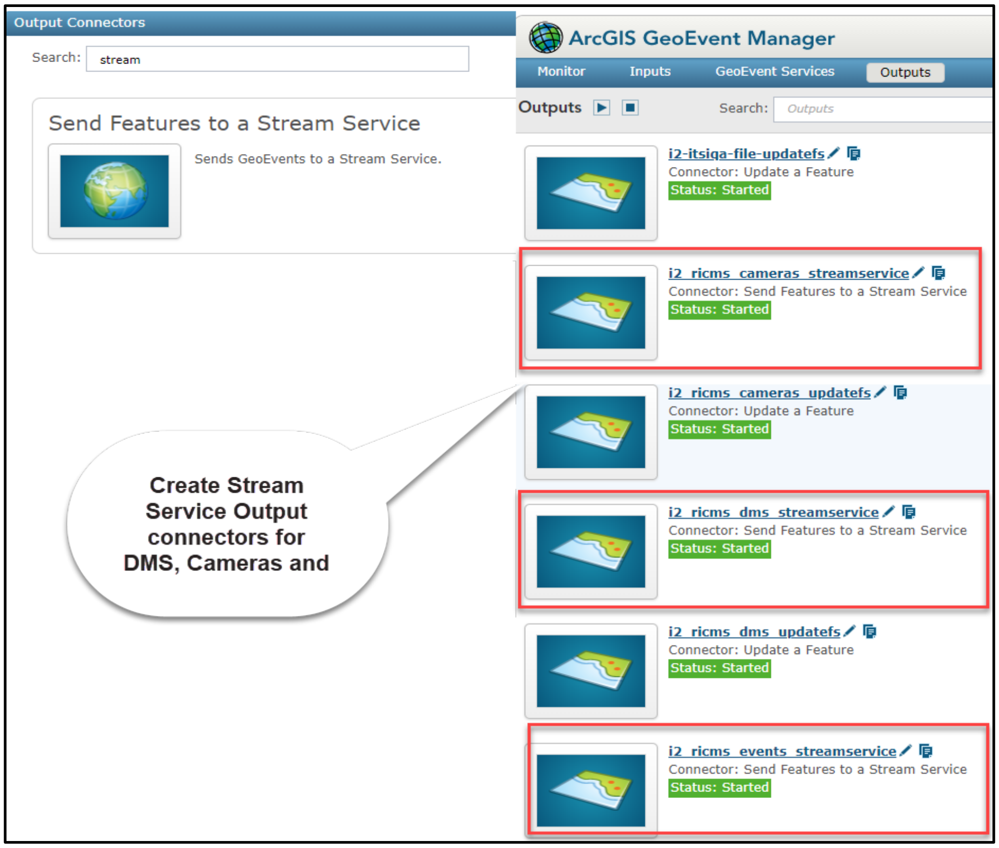
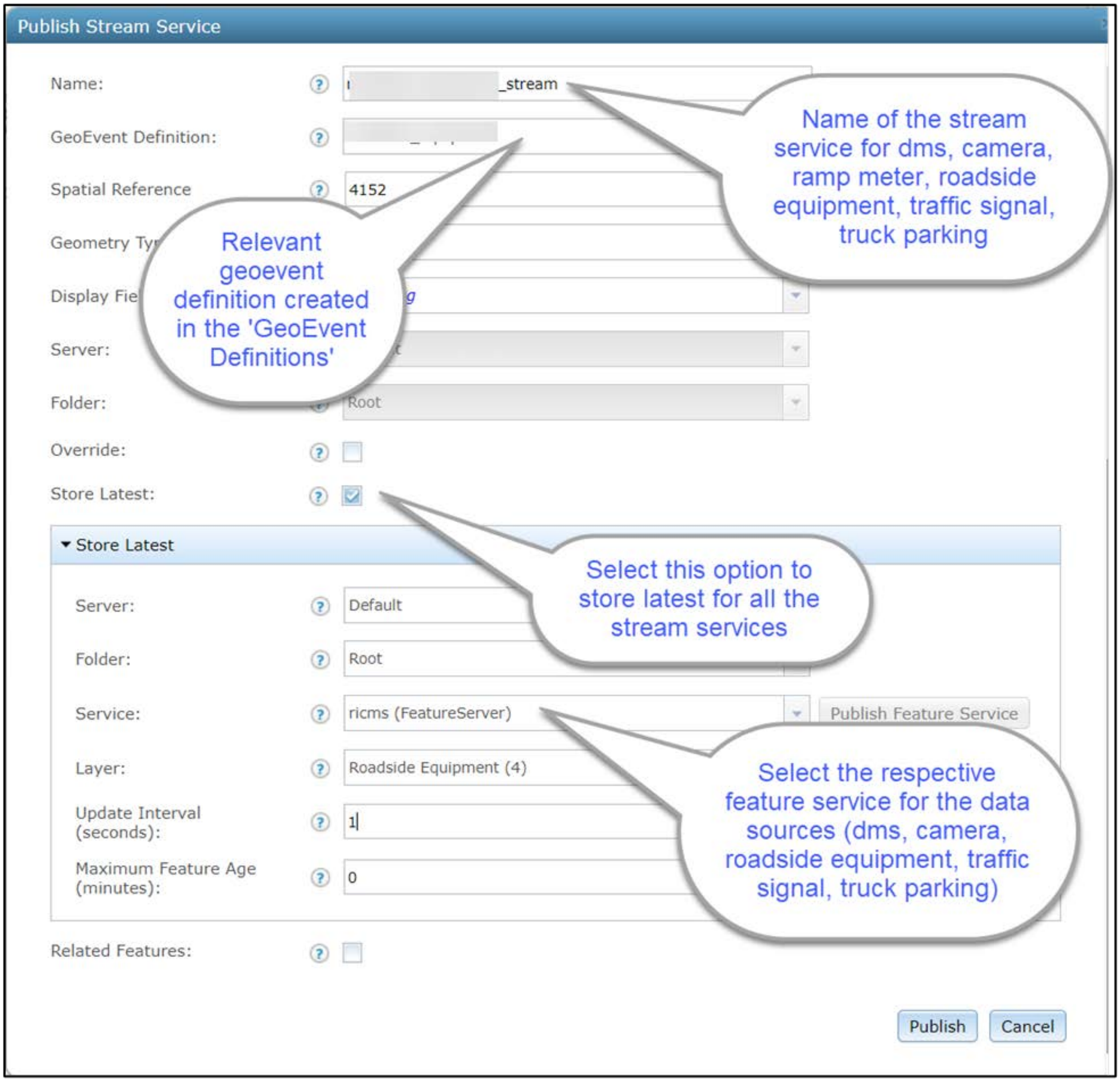

Output Connectors
Search: stream
Send Features to a Stream Service
Sends GeoEvents to a Stream Service.
ArcGIS GeoEvent Manager
Monitor
Inputs
GeoEvent Services
Outputs
Outputs
Search: Outputs
i2-itsiqa-file-updatefs
Connector: Update a Feature
Status: Started
i2_ricms_cameras_streamservice
Connector: Send Features to a Stream Service
Status: Started
i2_ricms_cameras_updatefs
Connector: Update a Feature
Status: Started
i2_ricms_dms_streamservice
Connector: Send Features to a Stream Service
Status: Started
i2_ricms_dms_updatefs
Connector: Update a Feature
Status: Started
i2_ricms_events_streamservice
Connector: Send Features to a Stream Service
Status: Started
Create Stream Service Output connectors for DMS, Cameras and

Figure 18 - Output Stream Service Connectors for DMS, Cameras, Events

4. While configuring the stream service output connectors, select the 'Store Latest' checkbox and refer the relevant feature service



Publish Stream Service

Name: [redacted]_stream

GeoEvent Definition: [redacted]

Spatial Reference: 4152

Geometry Ty

Display Fie g

Server: t

Folder: Root

Override: ☐

Store Latest: ☑

▾ Store Latest

Server: Default

Folder: Root

Service: ricms (FeatureServer)

Layer: Roadside Equipment (4)

Update Interval (seconds): 1

Maximum Feature Age (minutes): 0

Related Features: ☐

Name of the stream service for dms, camera, ramp meter, roadside equipment, traffic signal, truck parking

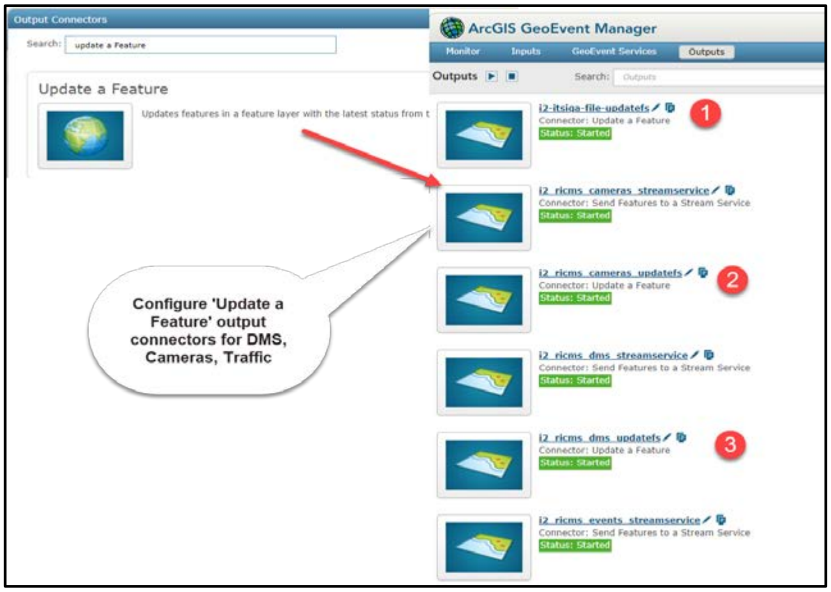
Relevant geoevent definition created in the 'GeoEvent Definitions'

Select this option to store latest for all the stream services

Select the respective feature service for the data sources (dms, camera, roadside equipment, traffic signal, truck parking)

Figure 19 – 'Store Latest' selection for stream services

5. Configure 'Update a Feature' output connector for DMS, Cameras, Ramp Meters, Roadside Equipment, Traffic Signals, Truck Parking, Vehicle Position and Traffic Conditions data sources



Search: update a Feature

Update a Feature

Updates features in a feature layer with the latest status from t

i2-itsiqa-file-updatefs
Connector: Update a Feature
Status: Started

i2_ricms_cameras_streamservice
Connector: Send Features to a Stream Service
Status: Started

i2_ricms_cameras_updatefs
Connector: Update a Feature
Status: Started

i2_ricms_dms_streamservice
Connector: Send Features to a Stream Service
Status: Started

i2_ricms_dms_updatefs
Connector: Update a Feature
Status: Started

i2_ricms_events_streamservice
Connector: Send Features to a Stream Service
Status: Started

Configure 'Update a Feature' output connectors for DMS, Cameras, Traffic

Figure 20 - 'Update a Feature' output connectors for DMS, Cameras, and Traffic data

6. Create GeoEvent Services connecting the relevant input and output connectors for all the dynamic data sources

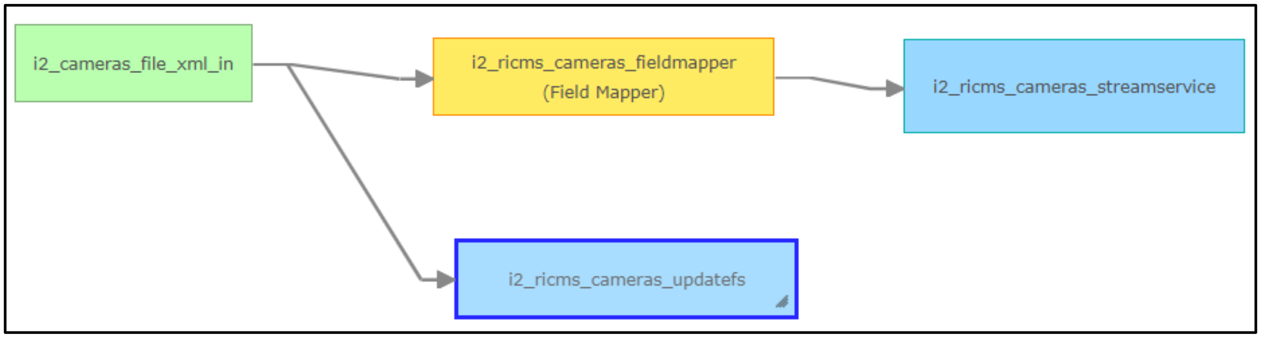
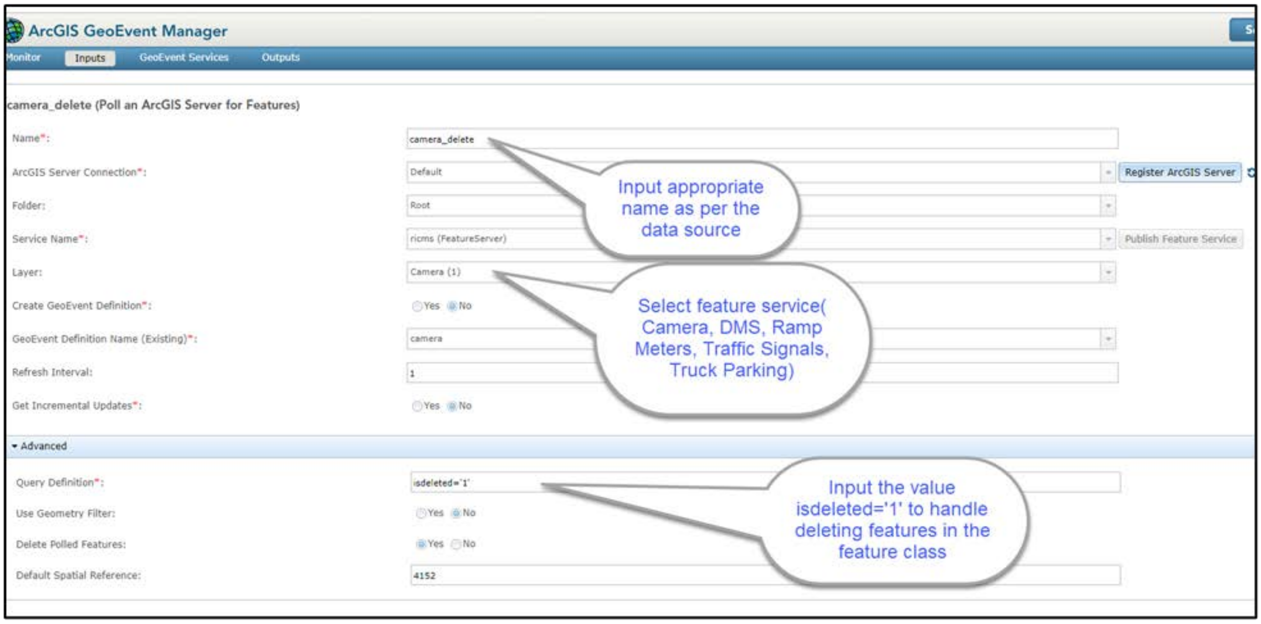

i2_cameras_file_xml_in
i2_ricms_cameras_fieldmapper
(Field Mapper)
i2_ricms_cameras_streamservice
i2_ricms_cameras_updatefs

Figure 21 - GeoEvent Service for Cameras

7. Create ‘Poll an ArcGIS Server for Features’ for Camera, DMS, Ramp Meters, Roadside Equipment, Traffic Signals, Truck Parking as below:



camera_delete (Poll an ArcGIS Server for Features)

Name*: camera_delete

ArcGIS Server Connection*: Default

Folder: Root

Service Name*: ricms (FeatureServer)

Layer: Camera (1)

Create GeoEvent Definition*: ○ Yes ● No

GeoEvent Definition Name (Existing)*: camera

Refresh Interval: 1

Get Incremental Updates*: ○ Yes ● No

Query Definition*: isdeleted='1'

Use Geometry Filter: ○ Yes ● No

Delete Polled Features: ● Yes ○ No

Default Spatial Reference: 4152

Input appropriate name as per the data source

Select feature service(Camera, DMS, Ramp Meters, Traffic Signals, Truck Parking)

Input the value isdeleted='1' to handle deleting features in the feature class

4.5.4 Signal Controller Log Powershell Script Installation

A powershell script must be installed and scheduled on the server where the signal controller logs are dropped for RICMS ingestion. This script is responsible for archiving many signal controller log files into a single zip archive to facilitate download by the signal controller log driver and to facilitate copying of the zip archive to multiple destination folders to support ingestion by more than one instance of the driver.

Installation and scheduling of the powershell script must be done as <USERNAME-AD-SERVICE-ACCOUNT>.

Follow these instructions to install and schedule the powershell task:

1. Copy create_zipfile.ps1 from Source\Drivers\SignalControllerLog\drop_folder_app to <PATH-ATSPM-DAT-DROP-PARENT>\ATSPM_DAT_DROP.
2. Copy SCLCreateZipFile.xml from Source\Drivers\SignalControllerLog\drop_folder_app\SCLCreateZipFile.xml to <PATH-ATSPM-DAT-DROP-PARENT>.
3. Log into <IP-ATSPM-DAT-DROP-SERVER> using Remote Desktop Connection
4. Launch Task Scheduler
5. Right-click on Task Scheduler Library and choose Import Task…
6. Navigate to the path on <IP-ATSPM-DAT-DROP-SERVER> where SCLCreateZipFile.xml was copied in step #3 and choose SCLCreateZipFile.xml and press Open.
7. In the Create Task window press OK to create the imported task.
8. Note the new folder named SCLCreateZipFile in the Task Scheduler Library and the new task within that folder.
9. Delete the local copy of SCLCreateZipFile.xml from <IP-ATSPM-DAT-DROP-SERVER>.

4.5.5 Orchestrator Installation

Two application orchestration tools are used to run R-ICMS: Kubernetes for the majority of applications, and Docker Swarm for applications that only run on Windows. Both orchestrators rely on Docker as the underlying runtime technology.

4.5.5.1 Docker Installation

Install Docker on all servers participating in either the Kubernetes cluster or Docker swarm. These servers are not a part of the Cloudera cluster, nor any of the database machines.

4.5.5.1.1On Ubuntu 18.04 LTS Machines

Install Docker by logging in as a sudoer and running the following commands from a terminal:

```
$ sudo apt-get update
$ sudo apt-get install -y \
  apt-transport-https \
  ca-certificates \
  curl \
  software-properties-common
$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-
key add –
$ sudo add-apt-repository \
  "deb [arch=amd64] https://download.docker.com/linux/ubuntu \
  $(lsb_release -cs) \
  stable"
$ sudo apt-get update
$ sudo apt-get install -y docker-ce=19.03.5-ce ~ce~3-0~ubuntu
```

On the Kubernetes servers:

```
$ sudo apt-get install -y docker-ce=17.03.3-ce ~ce~3-0~ubuntu
```

On the Docker registry server:

```
$ sudo apt-get install -y docker-ce=18.06.1-ce ~ce~3-0~ubuntu
```

Then create the file /etc/docker/daemon.json as root with the contents:

```
{
      "exec-opts": ["native.cgroupdriver=systemd"],
      "log-driver": "json-file",
      "log-opts": {
            "max-size": "100m"
      },
      "storage-driver": "overlay2",
      "insecure-registries" : ["image-repo.ricms:5000"]
}
```

Create and start a docker service:

```
$ sudo mkdir -p /etc/systemd/system/docker.service.d
$ sudo systemctl daemon-reload
$ sudo systemctl restart docker
```

4.5.5.1.2 Windows Authentication server

Install Docker 19.03.5 by running Powershell as an administrator and running the following commands:

Install Docker

```
PS> Install-Module DockerProvider -Force
```

If prompted to install NuGet provider, say yes.

```
Install NuGet provider: Y
PS> Install-Package Docker -ProviderName DockerProvider -Force -
RequiredVersion 19.03.5
```

Verify that docker is working

```
PS> docker images
```

Edit the docker daemon configuration file: C:\programdata\docker\config\daemon.json and add the following:

```
{
  "insecure-registries": ["<IMAGE-REPO-IP>", "image-repo.ricms:5000"]
}
```

Restart Windows

4.5.5.2 Kubernetes Installation and Setup

1. On all Kubernetes designated nodes, install all required Kubernetes packages by running the following commands as root:

```
$ sudo apt-mark unhold kubelet kubeadm kubectl
$ sudo apt-get remove kubelet kubeadm kubectl
$ sudo apt-get update
$ sudo apt-get install -y apt-transport-https curl
$ sudo apt-mark unhold kubelet kubeadm kubectl
$ sudo curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg |
  apt-key add –
  cat <<EOF >/etc/apt/sources.list.d/kubernetes.list
  deb https://apt.kubernetes.io/ kubernetes-xenial main
  EOF
$ sudo apt-get update
$ sudo apt-mark unhold kubelet kubeadm kubectl
$ sudo apt-add-repository "deb http://apt.kubernetes.io/ kubernetes-
  xenial main"
$ sudo apt-get update
$ sudo apt-get install -y kubelet kubeadm kubectl
$ sudo apt-mark hold kubelet kubeadm kubectl
```

2. On the Kubernetes <HOST-KUBERNETES-MASTER-0>, run the following commands:
    a. Create a file named kubeadm-config.yaml with the following contents:

```
apiVersion: kubeadm.k8s.io/v1beta2
kind: ClusterConfiguration
kubernetesVersion: "v1.17.0"
controlPlaneEndpoint: "10.32.92.40:6443"
networking:
      podSubnet: "10.244.0.0/16"
```

b. Run the following command in the directory where kubeadm-config.yaml is located:

```
$ sudo kubeadm init --config=kubeadm-config.yaml --upload-certs --ignore-preflight-errors=all
```

c. Copy the kube config to the user

```
$ cp /etc/kubernetes/admin.conf ~/.kube/config
```

d. Download the kube-flannel configuration file:

```
$ wget
https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml
```

e. Edit the `net-config.json` section of the `kube-flannel.yml` file as follows:

```
net-conf.json: |
  {
   "Network": "10.244.0.0/16",
   "Backend": {
    "Type": "vxlan",
    "VNI" : 4096,
    "Port": 4789
   }
  }
```

f. Edit the cni-conf.json section of the `kube-flannel.yml` file as follows:

```
cni-conf.json: |
  {
   "name": "vxlan0",
    "cniVersion": "0.2.0",
   "plugins": [
    {
     "type": "flannel",
     "delegate": {
      "hairpinMode": true,
      "isDefaultGateway": true
     }
    },
    {
     "type": "portmap",
     "capabilities": {
      "portMappings": true
     }
    }
   ]
  }
```

g. Apply the kube-flannel.yml

```
$ kubectl apply -f kube-flannel.yml
```

3. Perform the following actions on <HOST-KUBERNETES-MASTER-0> to retrieve information needed for steps 3 and 4

   a. Use the following command output for <TOKEN>:

   ```
   $ kubeadm token generate
   ```

   b. In the file /etc/Kubernetes/admin.conf, decode the base64 value of certificate-authority-data and use this value for <CA-CERT>.

   c. In the file /etc/Kubernetes/admin.conf, decode the base64 value of client-key-data and use this value for <KEY>.

4. On the remaining master nodes, run the following commands as root to join the cluster

   a. Reset kubeadm

   ```
   $ kubeadm reset
   ```

   b. Join the cluster, replace <HOST-KUBERNETES-MASTER-0> using the Placeholder Values table and the remaining placeholders with the values from step 3.

```
$ kubeadm join <HOST-KUBERNETES-MASTER-0>6443 --token <TOKEN>
--discovery-token-ca-cert-hash sha256:<CA-CERT> --control-plane
--certificate-key <KEY>
```

5. On all worker nodes, run the following commands as root to join the cluster
   a. Reset kubeadm

```
$ kubeadm reset
```

   b. Join the cluster, replace <HOST-KUBERNETES-MASTER-0> using the Placeholder Values table and the remaining placeholders with the values from step 3.

```
$ kubeadm join <MASTER-NODE-IP>:6443 --token <TOKEN> --discovery-
token-ca-cert-hash sha256:<CA-CERT>
```

4.5.5.3 Docker Image Registry Setup

On <HOST-KUBERNETES-MASTER-0>, ensure docker and docker-compose packages are installed.

Make the /mnt/registry directory if it does not exist: `sudo mkdir /mnt/registry`

Copy the source code folder Deploy/Registry to the server (user's home folder is fine).

Run the docker-compose command to build and run the registry:

```
$ sudo docker-compose -f ~/Registry/docker-compose.yml up -d --build
```

Add the registry image to the registry and verify using the registry API:

```
$ sudo docker push image-repo.ricms:5000/registry:latest
$ curl http://image-repo.ricms:5000/v2/_catalog
```

4.5.5.4 Docker Swarm Setup

4.5.5.4.1Manager Node Setup

On the Linux system that will be the Docker swarm manager, select one of its IP addresses that the worker nodes can access. Then run the following command as a sudoer on that machine:

```
$ sudo docker swarm init --advertise-addr 10.1.80.74
```

Copy the swarm join command from the output. On the Windows Server swarm worker, open Powershell as an Administrator, then paste and run the copied join command.

You will also need to open the following ports on the Windows Server swarm worker:

Table 8 - Windows Swarm Worker Ports

Port	Protocol	Reason
2377	TCP	Cluster management communications
7946	TCP and UDP	Communication among nodes
4789	UDP	Overlay network traffic

The status of the cluster can be confirmed by running the following command on the manager and ensuring all node statuses are "Ready":

```
$ sudo docker node ls
```

4.5.5.4.2 Windows Swarm Worker Node Setup

Verify all windows updates are installed before proceeding with the setup.

1. Run PowerShell as administrator, and run the following commands:
   ```
   PS> Install-WindowsFeature containers
   PS> Restart-Compute
   ```
2. Run PowerShell as administrator, and run the following commands:
   a. Verify OS build is later than 14393.1083 (required for swarm mode)
   ```
   PS> C:\Users\Administrator> systeminfo.exe | Select-String "Version"
   OS Version:          10.0.14393 N/A Build 14393
   BIOS Version:  Microsoft Corporation Hyper-V UEFI Release v1.0,
   11/26/2012
   ```
   b. Verify firewall is disabled and/or the required ports are open
   ```
   TCP port 2377 for cluster management communications
   TCP and UDP port 7946 for communication among nodes
   UDP port 4789 for overlay network traffic
   ```
   c. # open incoming ports
   ```
   PS> netsh advfirewall firewall add rule name="docker_cluster" dir=in
   action=allow protocol=TCP localport=2377
   PS> netsh advfirewall firewall add rule name="docker_swarm_tcp"
   dir=in action=allow protocol=TCP localport=7946
   PS> netsh advfirewall firewall add rule name="docker_swarm_udp"
   dir=in action=allow protocol=UDP localport=7946
   PS> netsh advfirewall firewall add rule name="docker_overlay" dir=in
   action=allow protocol=UDP localport=4789
   ```
   d. # open outgoing ports
   ```
   PS> netsh advfirewall firewall add rule name="docker_cluster"
   dir=out action=allow protocol=TCP localport=2377
   PS> netsh advfirewall firewall add rule name="docker_swarm_tcp"
   dir=out action=allow protocol=TCP localport=7946
   PS> netsh advfirewall firewall add rule name="docker_swarm_udp"
   dir=out action=allow protocol=UDP localport=7946
   PS> netsh advfirewall firewall add rule name="docker_overlay"
   dir=out action=allow protocol=UDP localport=4789
   ```
   e. # verify above opening the
   ```
   Control Panel (View: small icons) > Windows Firewall > Advanced
   Settings > (Inbound|Outbound) Rules
   ```
   or
   ```
   PS> netstat -an
   ```
3. Install Docker
   ```
   PS> Install-Module DockerProvider -Force
   ```
4. If prompted to install NuGet provider, say yes.
   ```
   Install NuGet provider: Y
   PS> Install-Package Docker -ProviderName DockerProvider -Force -
   RequiredVersion 18.09
   ```

5. Verify that docker is working
   ```
   PS> docker images
   ```
6. Install docker-compose (requires TLS 1.2):
   ```
   PS> [Net.ServicePointManager]::SecurityProtocol =
   [Net.SecurityProtocolType]::Tls12
   PS> Invoke-WebRequest
   "https://github.com/docker/compose/releases/download/1.24.0/docker-
   compose-Windows-x86_64.exe" -UseBasicParsing -OutFile
   $Env:ProgramFiles\Docker\docker-compose.exe
   ```
7. on Swarm manager computer, in terminal window, run
   ```
   > docker swarm join-token worker
   ```
8. Write down output command to run on new swarm nodes
9. In Windows Powershell, run the command output from the above “docker swarm join-token worker”.
   ```
   PS> docker swarm join --token SWMTKN-******** 10.1.80.74:2377
   ```

Add selection label to each node

10. On Swarm manager computer, in terminal window, run
    ```
    > docker node ls
    output:
    ```

```
ID                         HOSTNAME                  IP           OS
sif3rvl56y5s7cjifvyd7bz6a  RICMS-S16-2-DEV           10.1.80.78   Win 2016
ob84uvp194cbq8d6npz88bttc  RICMS-S16-3-DEV           10.1.80.84   Win 2016
y6vzxi0vk7t1pcbjyhc5rx0dq  RICMS-ST-AT-DEV           10.1.80.66   Win 2019
b7ed17bn89z1mazffk2qdqz9a  icms-docker-registry-dev  10.1.80.74   Ubun 16.04
```

Add label to each node based on its base OS:

```
> docker node update --label-add nodeid=win2016 sif3rvl56y5s7cjifvyd7bz6a
> docker node update --label-add nodeid=win2016 ob84uvp194cbq8d6npz88bttc
> docker node update --label-add nodeid=win2019 y6vzxi0vk7t1pcbjyhc5rx0dq
```

4.5.6 Application Installation and Deployment

To install and run the Kubernetes server applications for R-ICMS the applications that will run under Kubernetes must be built and pushed to a Docker image registry, the systems directed to use that registry, and the application manifests deployed. For the applications running under Docker swarm, similar setup is needed and additionally OpenFaaS should be deployed.

4.5.6.1 Deployment on Kubernetes

1. Prepare a build machine:
   - Requires a Windows 1809/2019 compatible operating system. The system version must match the base layer of Docker Windows images from mcr.microsoft.com in the source code, see:
     - Deploy/OpenFaas/HighwayCapacitySoftware/Dockerfile
     - Source/BusinessServices/Authentication/AuthenticationBusinessService/Dockerfile
     - Source/BusinessServices/Authorization/AuthorizationBusinessService/Dockerfile
   - Install Docker for Desktop using the link below:

https://docs.docker.com/docker-for-windows/install/

- Add insecure registry to "C:\ProgramData\Docker\config\daemon.json"
  ```
  "insecure-registries": [
     "image-repo.ricms:5000"
  ],
  ```
- Add registry host to "C:\Windows\System32\drivers\etc\hosts"
  ```
  <IMAGE-REPO-IP> image-repo.ricms
  ```
- Install kubectl using the link below:

  https://kubernetes.io/docs/tasks/tools/install-kubectl/#install-kubectl-on-windows
- Copy the .kube/admin.conf file from any Kubernetes master node into your local directory "$HOME\.kube\config.fdot ". Then run the command below and checking that the client and server versions match:
  ```
  $ kubectl –-kubeconfig="$HOME\.kube\<KUBE-CONFIG>" version
  ```

2. Configure source code

   a. Copy over a zip file of the source code to the build machine and unzip the file.

   b. Create an FDOT-D5 directory under Deploy\Kubernetes.

   c. Copy the contents of Deploy\Kubernetes\SwRI to the new FDOT-D5 directory.

   d. Configure the files in the new FDOT-D5 directory according to section 4.6.1.1.

3. Build and deploy the code

   a. Build the source code by running the script Deploy\Package-Deployment.ps1 with a named release tag (<RELEASE-TAG>) and the -Build option. Then verify the local docker cache contains images with matching IDs for the named release tag and 'latest' tag:
   ```
   $ Deploy\Package-Deployment.ps1 <RELEASE-TAG> <DEPLOY-ENV> -Build
   $ docker images
   ```

   b. Running the Pacakge-Deployment.ps1 script with the -Build options also creates Deploy\WindowsApiCaller\install.zip, containing the Windows API Caller Business Service, which is installed per section 4.5.6.3.

   c. Run the script Deploy\Package-Deployment.ps1 with the -Push option to push images to the registry host, and verify using the registry API:
   ```
   $ Deploy\Package-Deployment.ps1 <RELEASE-TAG> <DEPLOY-ENV> -Push
   $ curl http://image-repo.ricms:5000/v2/_catalog
   ```

   d. Optional: To optimize disk space, run the command below to clear the "image-repo.ricms:5000" registry of all but the latest images. Then verify the local docker cache contains only images with the named release tag and 'latest' tag:
   ```
   $ Deploy\Registry\Clean-Registry.ps1
   $ curl http://image-repo.ricms:5000/v2/_catalog
   ```

   e. If there are existing deployments in the Kubernetes cluster, run the script Deploy\Package-Deployment.ps1 with the -UnDeploy option.
   ```
   $ Deploy\Package-Deployment.ps1 <RELEASE-TAG> <DEPLOY-ENV> -Undeploy
   ```

   f. Verify there is nothing deployed to the cluster by running the following command:
   ```
   $ kubectl –-kubeconfig="$HOME\.kube\<KUBE-CONFIG>" get
     service,deployment,pods
   ```

   g. Ensure the SSL Certificates are located in the site specific deploy folder (*"Deploy\Kubernetes\<SITE>\certs"*).

h. Run the script Deploy\Package-Deployment.ps1 with the -Deploy option and a deployment target

```
$ Deploy\Package-Deployment.ps1 <RELEASE-TAG> <DEPLOY-ENV> -Deploy
```

i. It will take some time for the Kubernetes worker nodes to pull the Docker images and start the pods. The status of the deployment can be checked by running:

```
$ kubectl –-kubeconfig="$HOME\.kube\<KUBE-CONFIG>" get pods
```

j. The deployment is complete when all pods have a status of "**`Running`**".

Alternatively, if you have been provided access to ready-built images that have been pulled or unzipped and loaded into your repository, and you have been given the WindowApiCaller install.zip file, then you can skip building the images. In this case, you can perform steps 2a – d and steps 3c – i on a host with kubectl installed.

4.5.6.2 Deployment on Docker Swarm

Due to the large size of Windows Docker images, it is recommended to pull them manually before deploying the function in OpenFaas. Run the commands below on each Windows swarm worker:

```
PS> docker pull image-repo.ricms:5000/hcs7-faas-optimize:7.8-ltsc2019
```

```
PS> docker pull image-repo.ricms:5000/signal-controller-log-decode:latest
```

As a user with sudoer privileges on <HOST-SWARM-MANAGER-0>, copy the following RICMS source files into directory streets function in OpenFaas:

```
$ faas-cli deploy -f /opt/open-faas/functions/streets.yml
```

Deploy the signal log decoder function in OpenFaas:

```
$ faas-cli deploy -f /opt/open-faas/functions/signal-controller-log-decode.yml
```

Verify the function deployments:

```
$ sudo docker service ls
```

Function streets-optimize should have 4 replicas, and the function signal-controller-log-decode should have 10 replicas, values equal to the minimum replicas defined in their yml files.

4.5.6.3 Deployment of Auth using Task Scheduler and Windows Services

4.5.6.3.1 Windows Login API Caller

From the same machine used to build the source code in section 4.5.6.1, copy over the Windows Login API Caller install.zip to the designated machine and unzip it. Edit the file **`setup.ps1`** in a text editor, edit the following line and replace ***<HOST-KUBERNETES-WORKER-0>*** and ***<API-KEY>*** using the Placeholder Values table.

```
nssm-2.24\win64\nssm.exe set LoginApiCaller AppEnvironmentExtra
ASPNETCORE_URLS=http://+:5000 Redis:Endpoints:0:HostName=<HOST-
KUBERNETES-WORKER-0>
Redis:Endpoints:0:Port=6379 Authorization:ApiKey=<API-KEY>
```

Open PowerShell as an Administrator in that directory and run:

```
PS> .\setup.ps1
```

4.5.6.3.2 Authentication and Authorization Business Service

Copy the contents of the `Deploy\Docker` from the source code to the following location on the Windows Authentication Server: `"C:\Deploy\”`

Configure the `Deploy\Docker\auth-docker-swarm\docker-compose.yml` file according to section 4.6.1 and perform the following steps to configure a scheduled task:

1. Open Task Scheduler
2. Click “Create Task…” under the “Actions” section
3. In the “Name” text field, type the following: RICMS-Auth
4. In the “Description” text field, type the following: Starts the Authentication and Authorization Business Service for the RICMS on startup
5. Click the “Change User or Group…” button
6. Type “SYSTEM” in the text box and click “OK”
7. Click the checkbox that says “Run with highest privileges”
8. Click on the “Triggers” tab
9. Click “New…”
10. Click the “Begin the task” dropdown and select the “At startup” option
11. Select the “Delay task for” checkbox and set the value for 1 minute
12. Select the “Stop task if it runs longer than” checkbox and set the value for 1 hour
13. Click “OK”
14. Click on the “Actions” tab
15. Click “New”
16. In the “Program\script” text field, type the following path:
    `C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe`
17. In the “Add arguments” text field, type the following:
    `-ExecutionPolicy Bypass -File "C:\Deploy\Start-Auth.ps1"`
18. Click “OK”
19. Click “OK” again to save the task

The new scheduled task is now configured to start Authentication and Authorization Business services on start up of the machine. To initially start these services, open a PowerShell terminal, navigate to the Deploy folder and run the following command:

```
PS> .\Start-Auth.ps1
```

After the script completes, verify that the services have started by running the following command:

```
PS> docker ps
```

4.5.7 Power BI Report Server Installation and Deployment

This section does not replace the detailed instructions published by Microsoft for the installation of Power BI Report Server. These instructions only describe the highlights and are intended for persons experienced with installing Power BI Report Server products.

4.5.7.1 Install Power BI Report Server, Configure certificate

Download the PowerBIReportServer.exe from URL:

https://www.microsoft.com/en-us/download/details.aspx?id=57270

General Installation for Power BI Report Server are provided at this URL:

https://docs.microsoft.com/en-us/power-bi/report-server/install-report-server

- Review installation requirements, system configuration checks, and security considerations for a Power BI Report Server installation.

Verify a Reporting Services Installation are provided at this URL:

https://docs.microsoft.com/en-us/sql/reporting-services/install-windows/verify-a-reporting-services-installation?view=sql-server-ver15

Configure Power BI Report Server URL with SSL certificate are provided at this URL:

https://www.sqlservercentral.com/articles/configure-ssrs-with-an-ssl-certificate

Detailed instructions can be found at the following URL:

https://docs.microsoft.com/en-us/sql/reporting-services/install-windows/configure-a-url-ssrs-configuration-manager?view=sql-server-ver15

4.5.7.2 Configure Service Account

Configure the Report Server Service Account (Report Server Configuration Manager) are provided at this URL:

https://docs.microsoft.com/en-us/sql/reporting-services/install-windows/configure-the-report-server-service-account-ssrs-configuration-manager?view=sql-server-ver15

Name	Key	Value	Description
<<Power BI Report Server Service Account User Name>>	Username	<USERNAME-PowerBI-SERVICE-ACCOUNT>	The username to be used to configure Power BI Report Server Service account
<<Power BI Report Server Service Account Password>>	Password	<PASSWORD-PowerBI-SERVICE-ACCOUNT>	The Password to be used to configure Power BI Report Server Service account

4.5.7.3 Create PowerBI Report and Upload/Save

PowerBI reports can be created using the instructions at the following URL:

https://docs.microsoft.com/en-us/power-bi/report-server/quickstart-create-powerbi-report

Instructions on how to upload a file or report to the report server are provided at the following URL:

https://docs.microsoft.com/en-us/sql/reporting-services/reports/upload-a-file-or-report-report-manager?view=sql-server-ver15

4.5.7.4 Add Domain Users Security group to each Power BI Reports

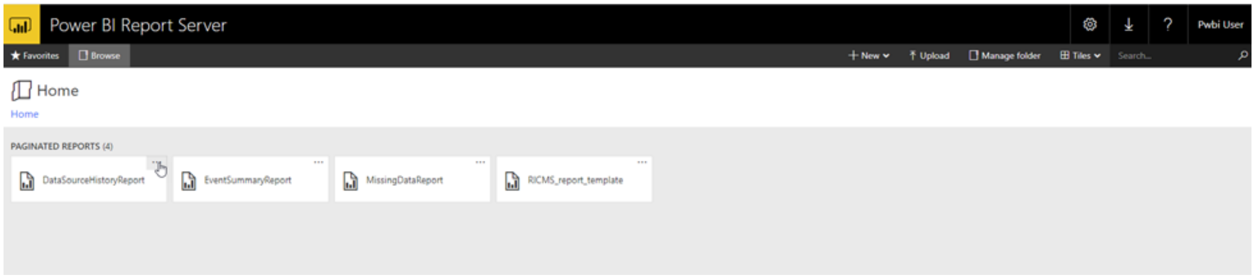
After you upload/save your report in Power BI Report Server, you manage security for each uploaded reports on the Power BI Web Portal. We need to perform this step so that every RICMS user can run the reports using their login credentials.

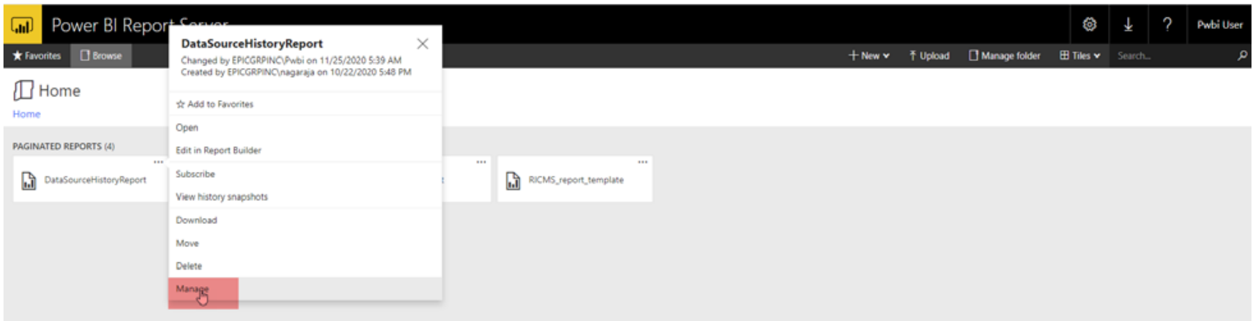
If the uploaded report does not have the roles it needs, you need to open the Power BI Report Server We Portal URL, add group or user, and then save it back to report.

Manage items in the web portal:

Power BI Report Server offers detailed control of the items you store on the web portal. For example, you can set up subscriptions, caching, snapshots, and security on individual paginated reports.

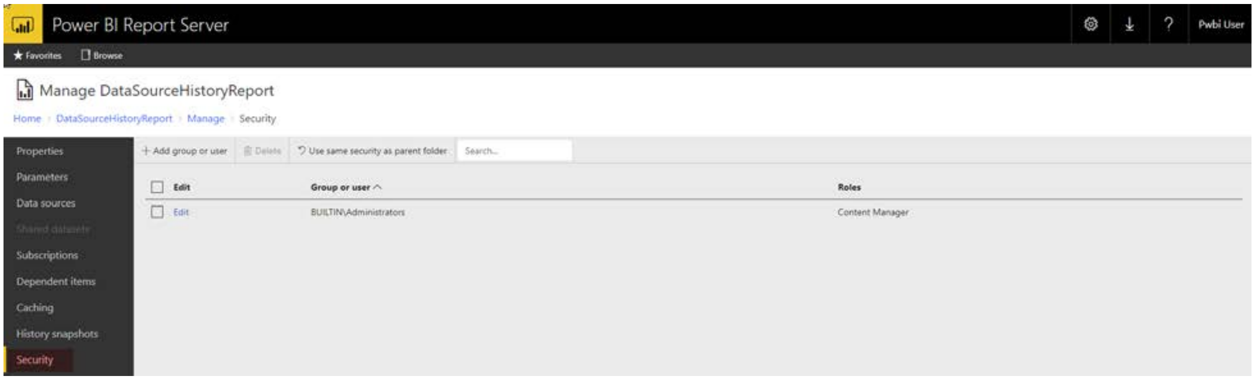
1. Select More options (...) in the upper-right corner of an item, then select Manage.





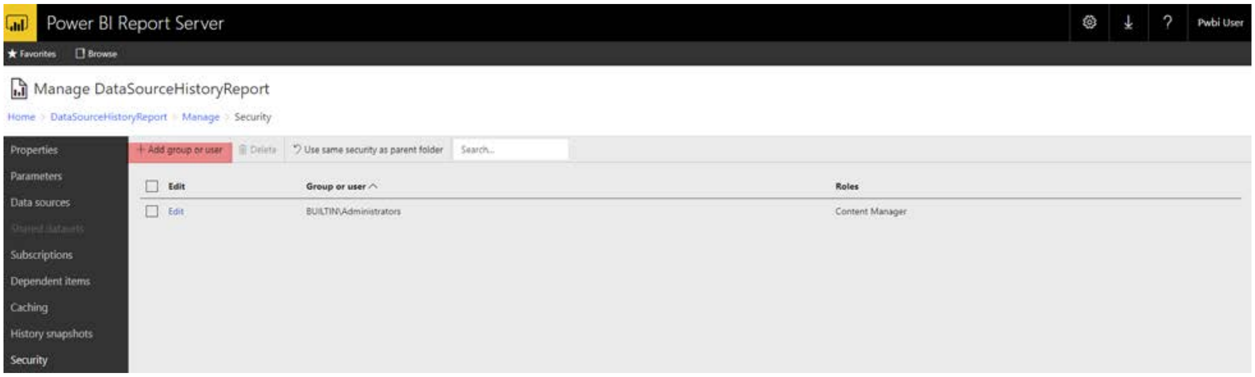
Changed by EPICGRPINC\Pwbi on 11/25/2020 5:39 AM
Created by EPICGRPINC\nagaraja on 10/22/2020 5:48 PM

2. Choose the property or other feature you want to set. Select **Security**



□ Edit	Group or user	Roles
□ Edit	BUILTIN\Administrators	Content Manager

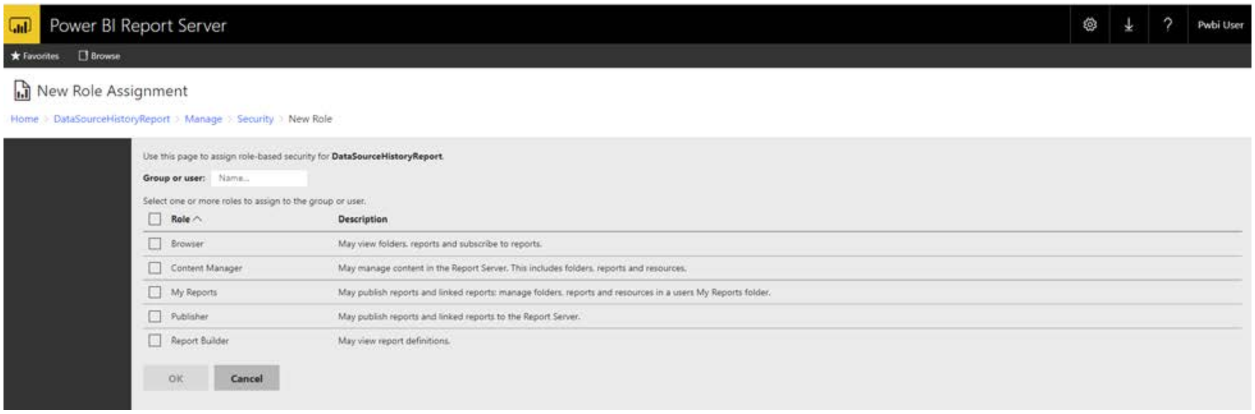
3. Click on **+ Add group or user**



Manage DataSourceHistoryReport

Home > DataSourceHistoryReport > Manage > Security

☐ Edit	Group or user ^	Roles
☐ Edit	BUILTIN\Administrators	Content Manager



New Role Assignment

Home > DataSourceHistoryReport > Manage > Security > New Role

Use this page to assign role-based security for **DataSourceHistoryReport**.

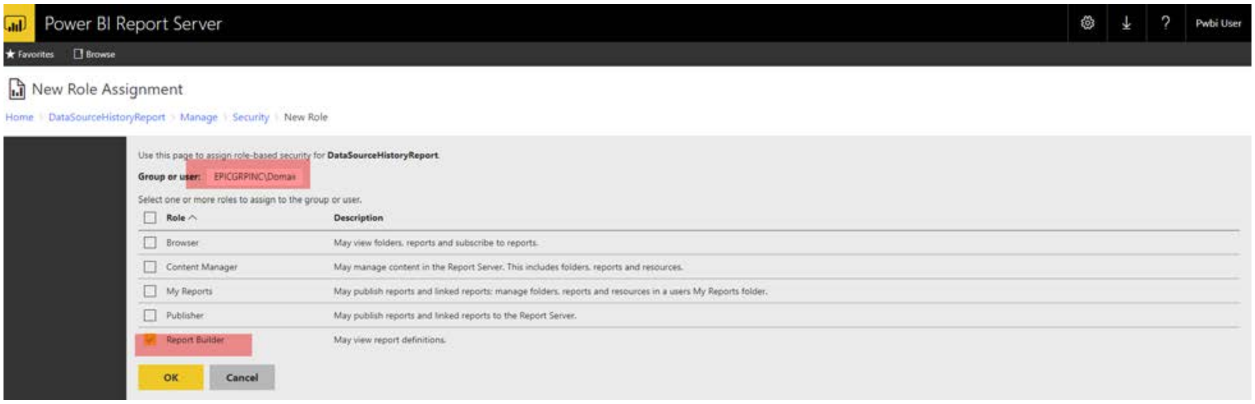
Group or user: Name...

Select one or more roles to assign to the group or user.

☐ Role ^	Description
☐ Browser	May view folders, reports and subscribe to reports.
☐ Content Manager	May manage content in the Report Server. This includes folders, reports and resources.
☐ My Reports	May publish reports and linked reports: manage folders, reports and resources in a users My Reports folder.
☐ Publisher	May publish reports and linked reports to the Report Server.
☐ Report Builder	May view report definitions.

4. Enter Group or User as DomainName\Domain Users, For example: D5-ITS\Domain Users

   Select Report Builder role



New Role Assignment

Home › DataSourceHistoryReport › Manage › Security › New Role

Use this page to assign role-based security for **DataSourceHistoryReport**.

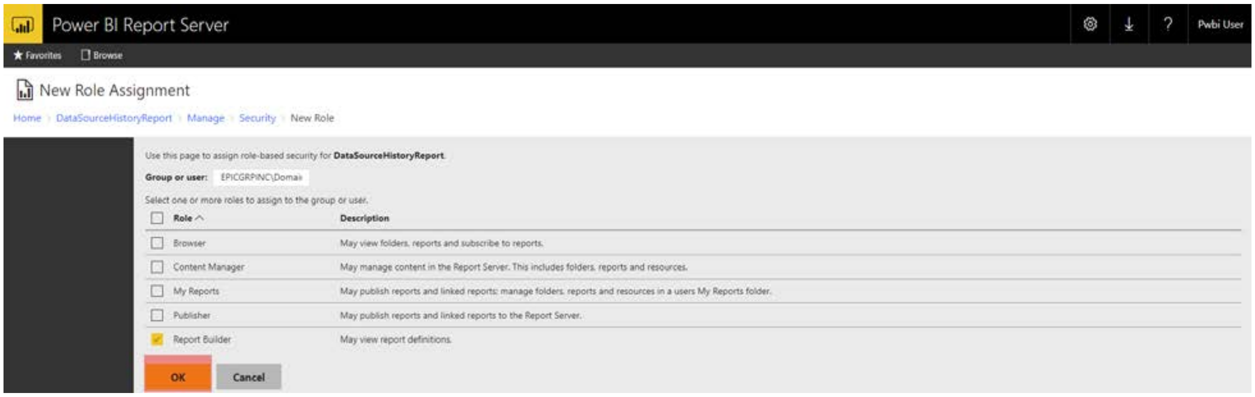
Group or user: EPICGRPINC\Domai

Select one or more roles to assign to the group or user.

	Role	Description
□	Browser	May view folders, reports and subscribe to reports.
□	Content Manager	May manage content in the Report Server. This includes folders, reports and resources.
□	My Reports	May publish reports and linked reports; manage folders, reports and resources in a users My Reports folder.
□	Publisher	May publish reports and linked reports to the Report Server.
☑	Report Builder	May view report definitions.

OK Cancel

5. Click Ok



New Role Assignment

Home › DataSourceHistoryReport › Manage › Security › New Role

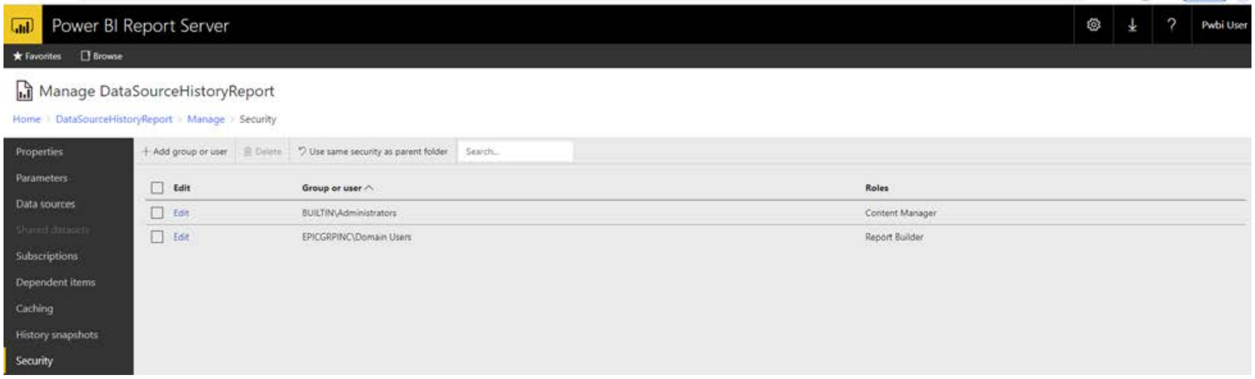
Use this page to assign role-based security for **DataSourceHistoryReport**.

Group or user: EPICGRPINC\Domai

Select one or more roles to assign to the group or user.

	Role	Description
□	Browser	May view folders, reports and subscribe to reports.
□	Content Manager	May manage content in the Report Server. This includes folders, reports and resources.
□	My Reports	May publish reports and linked reports; manage folders, reports and resources in a users My Reports folder.
□	Publisher	May publish reports and linked reports to the Report Server.
☑	Report Builder	May view report definitions.

OK Cancel



Manage DataSourceHistoryReport

Home › DataSourceHistoryReport › Manage › Security

	Edit	Group or user	Roles
□	Edit	BUILTIN\Administrators	Content Manager
□	Edit	EPICGRPINC\Domain Users	Report Builder

4.6 Configuration

The R-ICMS software has a number of configurable parameters that are used to establish communication between R-ICMS processes as well as to alter the behavior of the R-ICMS applications. The following sections describe the configuration parameters required for different site environments (i.e., SwRI, FDOT).

4.6.1 R-ICMS Applications and Services

4.6.1.1 Kubernetes Manifest Configuration

RICMS Kubernetes configuration takes place in two different directories. The first is the Common folder which contains base level configuration that should be the same for any deployment regardless of the environment. The second is the site-specific folder which needs to be configured for each environment. Configure the following values in the site-specific folder located in Deploy\Kubernetes\<SITE>:

Table 9 - Kubernetes Manifest Configuration

File Name: api-keys.yaml			
Section	**Key**	**Description**	**Base64 Encoded**
kind: Secret name: api-keys	<SERVICE_NAME>	The API Key used by a service for authorization. Each service that sends requests to another service will need a key/value.	Yes
File Name: authentication.yaml			
Section	**Key**	**Description**	**Base64 Encoded**
kind: Endpoints name: authentication-business-service	ip	The IP address of the machine hosting the Authentication Business Service.	No
kind: Endpoints name: authentication-business-service	port	The port number that the Authentication Business Service is listening on.	No
File Name: authorization.yaml			
Section	**Key**	**Description**	**Base64 Encoded**
kind: Secret name: authorization-database	database-connection-string	The connection string for the Authorization Database.	Yes
kind: ConfigMap name: authorization-data-service	appsettings.json	The appsettings configuration for the application.	No
kind: ConfigMap name: authorization-data-service	appsettings.json (IcmGroupGuid)	The Guid of the base level ICM group in Active Directory.	No

kind: ConfigMap name: authorization-data-service	appsettings.json (IcmGroupDN)	The Distinguished Name of the base level ICM group in Active Directory.	No
kind: ConfigMap name: authorization-data-service	appsettings.json (AdminGroupGuid)	The Guid of the Admin group in Active Directory.	No
kind: ConfigMap name: authorization-data-service	appsettings.json (AdminGroupDN)	The Distinguished Name of the Admin group in Active Directory.	No
kind: Endpoints name: authorization-business-service	ip	The IP address of the machine hosting the Authorization Business Service.	No
kind: Endpoints name: authorization-business-service	port	The port number that the Authorization Business Service is listening on.	No
File Name: data-quality-service.yaml			
Section	**Key**	**Description**	**Base64 Encoded**
kind: Secret name: data-quality-service-connectionstrings	data-quality-db	The connection string for the Data Quality Service Database.	Yes
File Name: divas.yaml			
Section	**Key**	**Description**	**Base64 Encoded**
kind: ConfigMap name: divas	uri	The connection URL to DIVAS.	No
kind: ConfigMap name: divas	username	The username for DIVAS.	No
kind: Secret name: divas	password	The password for DIVAS.	Yes
File Name: elastic-stack.yaml			
Section	**Key**	**Description**	**Base64 Encoded**
kind: ConfigMap name: kibana	kibana.yaml (elasticsearch.hosts)	An array of the Elasticsearch host addresses (“http://<IP>:<PORT>”).	No
kind: Service name: kibana	externalIPs	An array of IP address of the Kubernetes nodes.	No
kind: Secret name: kibana	password	The password for the ‘kibana’ user in Elasticsearch.	Yes
kind: Secret name: elastic	password	The password for the ‘elastic’ user in Elasticsearch	Yes
kind: ConfigMap name: heartbeat-config	elasticsearchHosts	Comma separated array of Elasticsearch host addresses.	No
kind: ConfigMap name: heartbeat-config	hosts	The host addresses of the applications to perform a heartbeat monitor on.	No
File Name: event-business-service.yaml			

Section	Key	Description	Base64 Encoded
kind: Secret name: event-business-service-connectionstrings	eventdb	The connection string for the Event Database.	Yes
kind: ConfigMap name: event-business-service	appsettings.json (Email.Cc)	CC recipient address for emails sent by the Event Data Service.	No
kind: ConfigMap name: event-business-service	appsettings.json (Kafka. BootstrapServers)	A comma separated array of the Kafka hosts (<IP>:<PORT>).	No
File Name: external-data-service.yaml			
Section	**Key**	**Description**	**Base64 Encoded**
kind: ConfigMap name: external-data-service	appsettings.json (GisServer)	The connection URL for the GIS Server.	No
File Name: fdot-ad.yaml			
Section	**Key**	**Description**	**Base64 Encoded**
kind: ConfigMap name: fdot-ad	username	Username for the FDOT Active Directory account.	No
kind: Secret name: fdot-ad	password	Password for the FDOT Active Directory account.	Yes
File Name: filebeat.yaml			
Section	**Key**	**Description**	**Base64 Encoded**
kind: DaemonSet name: filebeat	env	An array of environment variables in the following format: - name: ELASTICSEARCH_HOST1 value: <HOST1_IP>	No
kind: Secret name: filebeat	password	The password for the ‘elastic’ user in Elasticsearch	yes
File Name: gateway.yaml			
Section	**Key**	**Description**	**Base64 Encoded**
kind: Service name: gateway-nginx	externalIPs	An array of IPs of the Kubernetes worker nodes.	No
File Name: geoevent-share.yaml			
Section	**Key**	**Description**	**Base64 Encoded**
kind: ConfigMap name: geoevent-share	geoevent_share.json (remote_name)	The hostname of the server that hosts the GeoEvent shared folder.	No

kind: ConfigMap name: geoevent-share	geoevent_share.json (ip)	The IP address of the server that hosts the GeoEvent shared folder.	No
kind: ConfigMap name: geoevent-share	geoevent_share.json (port)	The port of the GeoEvent share folder.	No
kind: ConfigMap name: geoevent-share	geoevent_share.json (domain)	The full domain name of the server that hosts the GeoEvent shared folder. (Set to null if using a local username)	No
kind: ConfigMap name: geoevent-share	geoevent_share.json (service_name)	The shared folder name.	No
kind: ConfigMap name: geoevent-share	geoevent_share.json (geoevent_share_username)	The username to be used to authenticate to the GeoEvent share.	No
kind: Secret name: geoevent-share	password	The password to be used to authenticate the GeoEvent share.	Yes
File Name: gis-portal.yaml			
Section	**Key**	**Description**	**Base64 Encoded**
kind: ConfigMap name: gis-portal-credentials	username	The username for the GIS Portal account.	No
kind: Secret name: gis-portal-credentials	password	The password for the GIS Portal account.	Yes
File Name: gis-rest.yaml			
Section	**Key**	**Description**	**Base64 Encoded**
kind: ConfigMap name: gis-rest	map_server_url	The URL for the ARC GIS Rest services.	No
kind: ConfigMap name: gis-rest	map_server_query_params	A query parameter string for the map server URL.	No
kind: ConfigMap name: gis-rest	token_url	The URL for generating a token for the ARC GIS Rest services.	No
kind: ConfigMap name: gis-rest	token_referer	The base URL for the token generator.	No
File Name: hdfs.yaml			
Section	**Key**	**Description**	**Base64 Encoded**
kind: ConfigMap name: hdfs	hdfs.json (hdfs_hosts)	A comma separated array of the HDFS hosts.	No
kind: ConfigMap name: hdfs	hdfs.json (username)	The username for HDFS	No
kind: ConfigMap name: hdfs-connector-site-conf	site.properties (hadoop.fs.defaultFS)	The default Hadoop file system.	No

kind: ConfigMap name: hdfs-connector-site-conf	site.properties (hadoop.fs.name_node1)	The host address of Hadoop Node 1.	No
kind: ConfigMap name: hdfs-connector-site-conf	site.properties (hadoop.fs.name_node2)	The host address of Hadoop Node 2.	No
kind: ConfigMap name: hdfs-connector-site-conf	site.properties (hadoop.user_name)	The username for Hadoop.	No
kind: ConfigMap name: hdfs-connector-site-conf	site.properties (kafka.bootstrap.servers)	A comma separated array of the Kafka Boostrap hosts.	No
File Name: itsiqa-gap-data.yaml			
Section	**Key**	**Description**	**Section**
kind: Secret name: itsiqa-gap-mssql	uri	The connection string for the MSSQL Data Quality Database.	Yes
kind: Secret name: itsiqa-gap-mssql	server	The IP of the MSSQL Data Quality Database.	Yes
kind: Secret name: itsiqa-gap-mssql	user	The username for the MSSQL Data Quality Database.	Yes
kind: secret name: itsiqa-gap-mssql	password	The password for the MSSQL Data Quality Database.	Yes
File Name: kafka.yaml			
Section	**Key**	**Description**	**Base64 Encoded**
kind: ConfigMap name: kafka	kafka.json (boostrap_servers)	An array of Kafka Bootrap hosts.	No
kind: ConfigMap name: kafka	bootsrap-servers	A comma separated array of the Kafka Boostrap hosts.	No
File Name: metricbeat.yaml			
Section	**Key**	**Description**	**Base64 Encoded**
kind: ConfigMap name: metricbeat-daemonset-config	metricbeat.yml (setup.kibana.host)	The IP and Port of the Kibana host (<IP>:<PORT>)	No
kind: Daemonset name: metricbeat	env	An array of environment variables (ELASTICSEARCH_HOST<#>) and Elasticsearch IPs	No
kind: ConfigMap name: metricbeat-deployment-config	metricbeat.yml (setup.kibana.host)	The IP and Port of the Kibana host (<IP>:<PORT>)	No
kind: Deployment name: metricbeat	env	An array of environment variables (ELASTICSEARCH_HOST<#>) and Elasticsearch IPs	No
File Name: mongo.yaml			

Section	Key	Description	Base64 Encoded
kind: Secret name: mongo-database	uri	The connection string to the Mongo Database	Yes
kind: Secret name: mongo-database-readonly	uri	The read-only connection string to the Mongo Database	Yes
File Name: monitoring.yaml			
Section	**Key**	**Description**	**Base64 Encoded**
kind: Secret name: monitoring-database	database-connection-string	The connection string to the Monitoring Database	Yes
File Name: mssql-server.yaml			
Section	**Key**	**Description**	**Base64 Encoded**
kind: Secret name: mssql-server-database	connection-string	The connection string to the MSSQL Server.	Yes
File Name: notification.yaml			
Section	**Key**	**Description**	**Base64 Encoded**
kind: Secret name: notification-database	database-connection-string	The connection string to the Notification Database	Yes
kind: ConfigMap name: notification-business-service	appsettings.json (Host)	The host name of the SMTP server.	No
kind: ConfigMap name: notification-business-service	appsettings.json (Port)	The port of the SMTP server.	No
kind: ConfigMap name: notification-business-service	appsettings.json (Username)	The username for the SMTP account.	No
kind: ConfigMap name: notification-business-service	appsettings.json (FromAddress)	The ‘From’ address for emails sent by the Notification Service.	No
kind: ConfigMap name: notification-business-service	appsettings.json (ICMWebsiteLink)	The IP address of the RICMS website (http://<IP>).	No
kind: Secret name: notification-email	password	The password for the SMTP account.	Yes
File Name: nws-weather-alerts.yaml			
Section	**Key**	**Description**	**Base64 Encoded**
kind: ConfigMap name: nws-weather-alerts	nws_weather_alerts.json (active_alerts_url)	The Active Weather Alerts URL.	no

kind: ConfigMap name: nws-weather-alerts	nws_weather_alerts.json (user_agent)	The user agent for retrieving active weather alerts.	no
File Name: origin-destination.yaml			
Section	**Key**	**Description**	**Base64 Encoded**
kind: Secret name: origin-destination-database	connection-string	The connection string to the Origin Destination Database	Yes
File Name: pyodbc.yaml			
Section	**Key**	**Description**	**Base64 Encoded**
kind: Secret name: pyodbc-event-db	uri	The connection string to the PYODBC MSSQL Database	Yes
File Name: redis.yaml			
Section	**Key**	**Description**	**Base64 Encoded**
kind: Service name: redis	externalIPs	The IPs of the Kubernetes worker nodes.	No
kind: Service name: redis	port	The port Redis runs on.	No
File Name: reports-service.yaml			
Section	**Key**	**Description**	**Base64 Encoded**
kind: Service name: reports-service-connectionstrings	reportsdb	The connection string for the Reports Database.	No
File Name: response-plan-selection.yaml			
Section	**Key**	**Description**	**Base64 Encoded**
kind: Secret name: response-plan-selection-service	response-plan-section-database	The connection string for the Response Plan Selection Database.	Yes
kind: ConfigMap name: response-plan-selection-service	appsettings.json (SimulationBaseUri-AimSun)	The host address for the Aimsun Simulation Engine (http://<IP>:<PORT>)	No
kind: ConfigMap name: response-plan-selection-service	appsettings.json (CallbackBaseUri)	The DNS Server Address of the RICMS.	No
File Name: scl-decode.yaml			
Section	**Key**	**Description**	**Base64 Encoded**
kind: ConfigMap name: scl-dat-csv-decoder	decode_function_url	The host address and route for the Signal Controller Log Decoder (http://<IP>:<PORT>/function/signal-controler-log-decode)	No
File Name: shapefile-share.yaml			

Section	Key	Description	Base64 Encoded
kind: ConfigMap name: shapefile-share	shapefile_share.json (remote_name)	The host name of the Shapefile Share.	No
kind: ConfigMap name: shapefile-share	shapefile_share.json (ip)	The IP of the Shapefile Share.	No
kind: ConfigMap name: shapefile-share	shapefile_share.json (port)	The port of the Shapefile Share.	No
kind: ConfigMap name: shapefile-share	shapefile_share.json (service_name)	The service name of the Shapefile Share.	No
File Name: siia.yaml			
Section	**Key**	**Description**	**Base64 Encoded**
kind: ConfigMap name: siia-credentials	username	The username for the SIIA account.	No
kind: Secret name: siia-credentials	password	The password for the SIIA account.	Yes
kind: ConfigMap name: siia	siia.json (base_url)	The host address and route for SIIA (http://<IP>:<PORT>/siia.api/api)	No
File Name: sot.yaml			
Section	**Key**	**Description**	**Base64 Encoded**
kind: Secret name: sot	database	The connection string for the SOT Database.	Yes
kind: Secret name: sot	hangfire-database	The connection string for the SOT Hangfire Database.	Yes
kind: ConfigMap name: sot	simulation-async-uri	The host address for running SOT simulations (http://<IP>:<PORT>)	No
kind: ConfigMap name: sot	simulation-callback-uri	The host address for simulation callback (http://<IP>)	No
kind: ConfigMap name: sot	openfaas-callback-uri	The host address for OpenFaas callback (http://<IP>)	No
kind: ConfigMap name: sot	timezone	The time zone the application runs in.	Yes
kind: Service name: openfaas	externalName	The hostname of the OpenFaas service.	No
File Name: sunguide-broker.yaml			
Section	**Key**	**Description**	**Base64 Encoded**
kind: ConfigMap name: sunguide-broker	appsettings.json (SunGuide.DatabusHost)	The IP of SunGuide Databus.	No
kind: ConfigMap	appsettings.json	The port of SunGuide Databus.	No

name: sunguide-broker	(SunGuide.DatabusPort)		
File Name: sunguide.yaml			
Section	**Key**	**Description**	**Base64 Encoded**
kind: ConfigMap name: sunguide-credentials	username	The username for the SunGuide account.	No
kind: Secret name: sunguide-credentials	password	The password for the SunGuide account.	Yes
kind: ConfigMap name: sunguide-<SOURCE>	host	The host name for SunGuide.	No
File Name: user-interface.yaml			
Section	**Key**	**Description**	**Base64 Encoded**
kind: ConfigMap name: user-interface	config.json (reportServerBaseUrl)	The base URL of the Reports Server.	No
kind: ConfigMap name: user-interface	config.json (cc)	The 'CC' address to use when sending emails.	No
kind: ConfigMap name: user-interface	config.json (appurl)	The DNS Server address for the RICMS.	No
kind: ConfigMap name: user-interface	config.json (gisGenerateTokenUrl)	The URL for generating a token for the ARC GIS Rest services.	No
kind: ConfigMap name: user-interface	config.json (gisReferenceUrl)	The base URL for the token generator.	No
kind: ConfigMap name: user-interface	config.json (gisTokenUserName)	The username for generating a token for the ARC GIS Rest services.	No
kind: ConfigMap name: user-interface	config.json (gisTokenPassword)	The password for generating a token for the ARC GIS Rest services.	No
kind: ConfigMap name: user-interface	config.json (gisStreamServerUrl)	The URL of the GIS Stream Server.	No
kind: ConfigMap name: user-interface	config.json (gisFeatureServerUrl)	The URL of the GIS Feature Server	No
kind: ConfigMap name: user-interface	config.json (gisWeatherWarningUrl)	The URL of the GIS Weather Warning Server.	No
kind: ConfigMap name: user-interface	config.json (gisWeatherStormUrl)	The URL of the GIS Weather Storm Server.	No
kind: ConfigMap name: user-interface	config.json (gisPrecipitationUrl)	The URL of the GIS Precipitation Server.	No
kind: ConfigMap name: user-interface	config.json (mapInfo, for entry "FDOT D5 Basemap")	The URL for the basemap.	No
kind: ConfigMap name: user-interface	config.json (mapInfo, for entry "Response Plan")	The URL for Response Plan related GIS data.	No

Values identified with Encoded = Yes in the table above must be Base64 encoded. This does not encrypt the data but makes it non-obvious and difficult to memorize at a glance compared to plain text. To Base64 encode a value, run the following command on one of the Linux machines:

```
$ echo -n '<VALUE-TO-ENCODE>' | base64 -w 0; echo
```

To view the contents of a Base64 encoded value, run the following command:

```
$ echo <VALUE-TO-DECODE> | base64 --decode; echo
```

4.6.1.2 Windows Auth Services Configuration

The Windows Auth services are configured using a docker-compose file located at Deploy\Docker\auth-docker-swarm\docker-compose-yml. This file is to be used as a template for configuration for a new environment. Using the following table, replace all the template placeholders in the file with the site-specific values:

Table 10 - Windows Authorization Service Configuration

Template Placeholder	Description
<API_KEY>	The API key that is used for authorization against other services
<REDIS_HOST>	The IP address where Redis is running. This should be the IP of one of the Kubernetes nodes.
<REDIS_PORT>	The port that Redis is listening on. This should be 6379 unless otherwise specified.
<LOGIN_API_CALLER_CONNECTION_STRING>	The host address and route to the Windows Login API Caller.
<AUTH_DATA_CLIENT_CONNECTION_STRING>	The host address and route to the Authorization Data Service. The base address should be the IP of one of the Kubernetes nodes. The route should be /api/authorization-data-service.
<HANGFIRE_CONNECTION_STRING>	The connection string for the Hangfire Database.
<MAIN_DOMAIN>	The Active Directory domain for the RICMS users.

<MAIN_DOMAIN_SEARCH_BASE>	The Active Directory search base for the domain.
<ICM_GROUP_NAME>	The name of the Active Directory group associated with RICMS users.
<GROUP_DISTINGUISHED_NAME>	The Active Directory distinguished name of the RICMS group.
<HOST_NAME>	The IP address of the LDAP host.
<DOMAIN>	The domain of the LDAP host.
<PORT>	The port of the LDAP host.
<USE_SSL>	A boolean to determine if SSL will be used or not.
<SEARCH_BASE>	The search base for the domain of the LDAP host.
<USERNAME>	The LDAP service account username.
<PASSWORD>	The LDAP service account password.

4.6.1.3 Elasticsearch Watchers

The Elasticsearch Watchers require site-specific configuration. The site-specific configuration for the Elasticsearch Watchers is located at ***`Deploy\Monitoring\Watchers\<SITE>`***. If there is not a folder for the designated site, copy an existing one and rename the folder. Navigate into the site folder and edit the ***`settings.json`*** file. Replace the values of the following keys:

Key	**Description**
elasticsearchHost	The host name of one of the Elasticsearch nodes.
elasticsearchPassword	The password for the default 'elastic' user in Elasticsearch (base64 encoded).
kubernetesHost	The host name of one of the Kubernetes nodes.
apiKey	The API key used by the Watcher to send requests to the Monitoring Service and the Data Quality Servicer (base64 encoded).

cpuUtilization	A percentage of CPU utilization that meets or exceeds this value will result in an alert.
diskSpaceUtilization	A percentage of Disk Space utilization that meets or exceeds this value will result in an alert.
ramUtilization	A percentage of RAM utilization that meets or exceeds this value will result in an alert.
retrievalIntervals (driverName, dataGapThreshold, alertThreshold)	An array of drivers and their expected data retrieval intervals. The 'alertThreshold' is typically set to double that of the 'dataGapThreshold'.

4.6.1.4 Java Web Token API Keys

RICMS services use Java Web Tokens (JWT) for authorization. Initially, there will only be 1 valid JWT that is created as a part of the initial database seeding. Use the following value when a configuration calls for an API key:

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VybmFtZSI6IkZET1QtRDUiLCJpc3MiOiJSSUNNUyIsImF1ZCI6IlJJQ01TIn0.7txOFG3WLQpve9PF0_ZjIh4wbRHz6CYLgCAqPPTShfk
```

This API key should only be used temporarily. Once the system is properly running, delete the existing API key and create a new one using the **api-key** route described in the source code file API\AuthorizationDataService.

4.6.2 Expand HDFS Capacity

The storage capacity of HDFS can be increased by adding a new worker node to the cluster. Perform below steps to add a worker node to the cluster.

1. Log into cloudera manager
    a. In browser open https://<HOST-CLOUDERA-UTILITY-FQDN-0>:7183
    b. Enter the credentials for the admin user
2. Note the original capacity of HDFS
    a. Click on the HDFS service from the Cloudera Manager home page
    b. The Configured Capacity is displayed in the HDFS Summary section of the page
3. Choose Hosts -> Add Hosts

4. Choose Add hosts to cluster and the cluster RICMS should already be chosen (since there is only one cluster) and press Continue
5. On the Setup Auto-TLS page, follow the instructions for “if you are using an existing Certificate Authority…”
    a. Log in to the Cloudera Manager server host
        i. ssh <USERNAME-AD-HUMAN-ACCOUNT>@<HOST-CLOUDERA-UTILITY-IP-0>
    b. Import the certificate and the key for each host…
        i. You must run the certmanager command as root (not simply with sudo)

        ```
        $ sudo su
        ```

        ii. Run the following command

        ```
        $ JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64
        /opt/cloudera/cm-agent/bin/certmanager --location
        /var/lib/cloudera-scm-server/certmanager add_custom_cert --
        host-cert <PATH_TO_WORKER_NODE_PEM_CERTIFICATE> --host-key
        <PATH_TO_WORKER_NODE_KEY> <WORKER_NODE_FQDN>
        ```

    c. In Cloudera manager UI press Continue
6. On the Specify Hosts page
    a. Type <WORKER_NODE_FQDN> and press the Search button
    b. Search results will appear below the Search button and the host to add will appear listed and selected/checked.
    c. Press Continue
7. On the Select Repository page press Continue
8. On the Accept JDK license page check the box to Install Oracle Java… and press Continue
9. On the Enter Login Credentials page
    a. Choose Login To All Hosts As: root
    b. Authentication Method: All hosts accept same password
    c. Enter Password: <CLOUDERA-ROOT-PASSWORD>
    d. Confirm Password: <CLOUDERA-ROOT-PASSWORD>
    e. Press continue
10. On the Install Agents page simply wait until the installation is complete and the wizard will automatically proceed to the Install Parcels page
    a. Note: Parcel installation may take a few minutes
11. After parcels are installed the wizard will automatically proceed to the Inspect Hosts for Correctness page. This step only takes about 30 seconds so let it run to completion. Once complete press Continue
12. On the Select Host Template page
    a. Choose None as host template and leave checkbox checked (by default) and press Continue
13. At this point the node has been added to the cluster but no roles have been configured for the new node. The node should be configured with the following roles like other existing worker nodes.
    a. HBase RegionServer
    b. HDFS DataNode
    c. Hive Gateway
    d. Impala Daemon

   e. Spark Gateway
   f. YARN (MR2 Included) NodeManager

14. To facilitate adding the above roles to this and future workers added to the cluster, RICMS added a host template named HadoopWorker. We will apply this host template.
    a. Go to Hosts -> All Hosts
    b. Check the checkbox next to <WORKER_NODE_FQDN>
    c. In the Actions for Selected (1) menu choose Apply Host Template
    d. In the Apply Host Template dialog
        i. Choose Host Template -> HadoopWorker (RICMS)
        ii. Check the Deploy client configurations and start newly created roles…
        iii. Press Continue
    e. Wait for the process to complete
15. Note the new capacity of HDFS
    a. Click on the HDFS service from the Cloudera Manager home page
    b. The Configured Capacity is displayed in the HDFS Summary section of the page

4.6.3 Remove Worker Node from Cloudera Cluster

To uninstall a node (i.e., after doing dry run testing prior to actual testing)

1. Stop the roles on the worker
    a. Go to Hosts -> All Hosts
    b. Check the checkbox next to <WORKER_NODE_FQDN>
    c. In the Actions for Selected (1) menu choose Stop Roles on Hosts
    d. Wait for roles to stop
    e. This may take a few minutes
2. Remove the worker from the cluster
    a. Go to Hosts -> All Hosts
    b. Check the checkbox next to <WORKER_NODE_FQDN>
    c. In the Actions for Selected (1) menu choose Remove Hosts From Cluster
    d. On the Remove Hosts From Cluster accept the defaults and press Confirm
    e. Wait for the node to be removed
    f. This may take a few minutes
3. Remove the worker from cloudera manager
    a. Go to Hosts -> All Hosts
    b. Check the checkbox next to <WORKER_NODE_FQDN>
    c. In the Actions for Selected (1) menu choose Remove Hosts from Cloudera Manager
    d. On the Remove Hosts From Cloudera Manager page it instructs you to first stop the Cloudera Manager agent on the host.

```
ssh <human account>@<WORKER_NODE_FQDN>
$ sudo service cloudera-scm-agent stop
```

    e. After the agent is stopped press Confirm to remove the host
    f. Go to Hosts -> All Hosts and observe that the host no longer appears in the list
4. Remove the cert directory that was created when adding the host

```
ssh <human account>@<WORKER_NODE_FQDN>
```

```
$ sudo rm -fr /var/lib/cloudera-scm-agent/agent-cert
```

4.6.4 Expand capacity of Logical Volume on MongoDB servers

1. FDOT MongoDB servers have no empty drive slots to increase storage capacity within the server.
2. However, the workaround is to add Direct Attached Storage (DAS) and extend the logical volume of mongo onto physical volume of DAS.
3. Currently we don’t have DAS hence in the below steps we demonstrate how to extend the size of logical volume onto free physical storage on the server.
4. Log into one of the MongoDB servers

   SSH <USERNAME-AD-HUMAN-ACCOUNT>@<MONGODB-SERVER>
5. Note the original size for /data/mongo

   ```
   $ sudo df –h /data/mongo
   ```

   Note the value in the size column.
6. Increase the size of the logical volume lv_data to use all of the available space in volume group vg_mongo

   ```
   $ sudo lvextend –n /dev/mapper/vg_mongo-lv_data + 100%FREE
   $ sudo      resize2fs /dev/mapper/vg_mongo-lv_data
   ```
7. Note the new size for /data/mongo

   ```
   $ sudo df –h /data/mongo
   ```

   Note the value in the size column. It should be greater than the previously recorded value.
8. Repeat the steps on other Mongo Servers in the cluster.

5 Notes

N/A