

CLAS

Configuration Location
Analytics Software



ZEBRA

Server and Software Installation Guide

Copyright

ZEBRA and the stylized Zebra head are trademarks of Zebra Technologies Corporation, registered in many jurisdictions worldwide. All other trademarks are the property of their respective owners. ©2020 Zebra Technologies Corporation and/or its affiliates. All rights reserved.

COPYRIGHTS & TRADEMARKS: For complete copyright and trademark information, go to www.zebra.com/copyright.

WARRANTY: For complete warranty information, go to www.zebra.com/warranty.

END USER LICENSE AGREEMENT: For complete EULA information, go to www.zebra.com/eula.

Terms of Use

- Proprietary Statement

This manual contains proprietary information of Zebra Technologies Corporation and its subsidiaries ("Zebra Technologies"). It is intended solely for the information and use of parties operating and maintaining the equipment described herein. Such proprietary information may not be used, reproduced, or disclosed to any other parties for any other purpose without the express, written permission of Zebra Technologies.

- Product Improvements

Continuous improvement of products is a policy of Zebra Technologies. All specifications and designs are subject to change without notice.

- Liability Disclaimer

Zebra Technologies takes steps to ensure that its published Engineering specifications and manuals are correct; however, errors do occur. Zebra Technologies reserves the right to correct any such errors and disclaims liability resulting therefrom.

- Limitation of Liability

In no event shall Zebra Technologies or anyone else involved in the creation, production, or delivery of the accompanying product (including hardware and software) be liable for any damages whatsoever (including, without limitation, consequential damages including loss of business profits, business interruption, or loss of business information) arising out of the use of, the results of use of, or inability to use such product, even if Zebra Technologies has been advised of the possibility of such damages. Some jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, so the above limitation or exclusion may not apply to you.

Revision History

Changes to the original guide are listed below:

Change	Date	Description
-01 Rev A	11/2018	Initial Release
-02 Rev A	4/2019	<p>Updates:</p> <ul style="list-style-type: none">- Configuring RTLS code.- Various URLs.- aar_info.csv file name.- Guide name change.- Chapter name changes to add CLAS.- Updated Installing Certificates.- Updated code under Configuring RTLS.- Replace Table 3 under Advanced Configuration Options.- Updated figures/diagrams. <p>Additions:</p> <ul style="list-style-type: none">- Standard About This Guide sections.- Local License Server Administrator Guide for Windows/Linux references.- Registry and legacy methods RTLS installs.- Preparing to run RTLS.- Added step to Registry Method.
-03 Rev A	2/2020	<p>Updates:</p> <ul style="list-style-type: none">- References to license information has been removed.- Docker registry address for CLAS download.- Instructions to start and stop CLAS.
-04 Rev A	6/2021	<p>Updates:</p> <p>Installing RTLS section. Topics added are:</p> <ul style="list-style-type: none">- Online Installation with Helper Script- Offline Installation

About This Guide

Introduction	6
Chapter Descriptions	6
Notational Conventions	6
Related Documents	7
Service Information	7
Provide Documentation Feedback	7

ZAATS Introduction

Introduction	8
Product Overview	8
Description and Features	8
ATR7000 Advanced Array Reader (AAR)	9
RTLS Services	10
RTLS Configuration and Management (CNM)	10
Location Analytics (LA)	11
Radio Control & Data (CND)	11
ZAATS Interfaces	11
REST Interface	12
Location Data Interface	12
Events Interface	12

Getting Started

Introduction	13
Overview	13
Hardware Requirements	13
Software Requirements	14
Registry Access	14
Installation Checklist	15

Installing CLAS Software

Introduction	17
CLAS Docker Overview	17
Installing RTLS Behind a Proxy Server	18
Installing RTLS	20
Online Installation with Helper Script	20
Installation Procedure	20
Online Installation without Helper Script	22
Installation Procedure	23
Offline Installation	24
Installation Procedure	25
Installing Certificates	26
Installing Certificates for Secure REST Interface	26
Installing Certificates for Secure Kafka Communication	27
Installing Certificates for Secure MQTT Communication	27
Changing the Password for the REST Interface	27
Configuring RTLS	29
Default contents rtls.conf file	29

Table of Contents

Basic Configuration Options	31
Advanced Configuration Options	32
Adding ATRs to RTLS	34
Preparing to Run RTLS	35
Starting and Stopping RTLS	35
Starting RTLS	35
Stopping RTLS	36
 Validating a CLAS and RTLS Services Installation	
Introduction	37
RTLS Simulator Architecture	37
Starting RTLS in Simulation Mode	38
Verification	39
Stopping the Simulation	39
 Running CLAS with ATR7000 Hardware	
Bringing the RTLS System Live with ATR7000 Readers	41

About This Guide

Introduction

This guide provides information about the requirements and configuration for a server hosting Configuration Location Analytics Software (CLAS) and Real-Time Location System (RTLS) software. It also describes the installation of RTLS software onto that server.

Chapter Descriptions

Topics covered in this guide are as follows:

- [ZAATS Introduction](#) provides an overview of Zebra's Advanced Asset Tracking System, including the ATR7000 overhead RFID readers, and the CLAS and RTLS Services software.
- [Getting Started](#) provides an overview of the steps required to install and validate a CLAS and RTLS Services software deployment, including hardware, software, and network related requirements.
- [Installing CLAS Software](#) describes the procedure to install and configure the various software components that comprise RTLS.
- [Validating a CLAS and RTLS Services Installation](#) provides information on the sequence of operations required to verify an RTLS Services software and server installation with the RTLS Simulator.
- [Running CLAS with ATR7000 Hardware](#) describes running RTLS with the ATR7000 hardware.

Notational Conventions

The following conventions are used in this document:

- The **Consolas** font is used throughout the guide to represent code.
- **Bold** text is used to highlight the following:
 - Dialog box, window and screen names
 - Drop-down list and list box names
 - Check box and radio button names
 - Icons on a screen
 - Key names on a keypad
 - Button names on a screen.

- Bullets (•) indicate:
 - Action items
 - Lists of alternatives
 - Lists of required steps that are not necessarily sequential.
- Sequential lists (for example, those that describe step-by-step procedures) appear as numbered lists.

Related Documents

The following documents provide more information about ZAATS:

- ZAATS Deployment Guide
- CLAS API Developer Guide
- ZAATS Tag and Numbering Guide
- ATR7000 Integration Guide

For the latest version of this guide and all guides, go to: www.zebra.com/support.

Service Information

If you have a problem with your equipment, contact Zebra Global Customer Support for your region. Contact information is available at: www.zebra.com/support.

When contacting support, please have the following information available:

- Serial number of the unit
- Model number or product name
- Software type and version number.

Zebra responds to calls by email, telephone or fax within the time limits set forth in support agreements.

If your problem cannot be solved by Zebra Customer Support, you may need to return your equipment for servicing and will be given specific directions. Zebra is not responsible for any damages incurred during shipment if the approved shipping container is not used. Shipping the units improperly can possibly void the warranty.

If you purchased your Zebra business product from a Zebra business partner, contact that business partner for support.

Provide Documentation Feedback

If you have comments, questions, or suggestions about this guide, send an email to EVM-Techdocs@zebra.com.

ZAATS Introduction

Introduction

Zebra's Advanced Asset Tracking System (ZAATS) provides continuous identification, location, and tracking of items tagged with passive UHF RFID tags conforming to the EPC™ Radio Frequency Identity Protocols Generation-2 UHF RFID Specification for RFID Air Interface standard. ZAATS is designed to enhance the efficiency and workflows of Zebra's customers' operations, which are increasingly focused on cohesive real-time data.

ZAATS consists of two primary components: the Real-Time Location System (RTLS) Services software, which contains the configuration, management, and location analytics components; and the ATR7000 overhead array readers.

Product Overview

Description and Features

ZAATS is a passive UHF RFID based asset tracking solution developed primarily for indoor warehousing, manufacturing, and logistics applications. It is based on the ATR7000 overhead RFID reader containing Zebra's proprietary advanced array architecture with integral antenna capable of steering beams and estimating the bearing to RFID tagged items with unprecedented accuracy and speed.

A summary of the key system and product features of ZAATS includes:

- A passive UHF RFID RTLS system that provides real-time identification and location data of tagged items for continuous asset monitoring.
- Configuration and Location Analytics Software (CLAS) that configures, controls, and manages the system, as well as a high-performance location analytics engine capable of providing up to 100,000 (1000 readers at 100 tps) tag location estimates per second with 2 ft typical accuracy.
- APIs to configure, manage, and control the ZAATS system using an HTTP-based RESTful interface.
- Docker container virtualization to simplify integration and deployment into end-user and partner applications.
- Software tools and documentation to facilitate system installation, including site planning, calibration, initial start-up, and deployment validation testing.

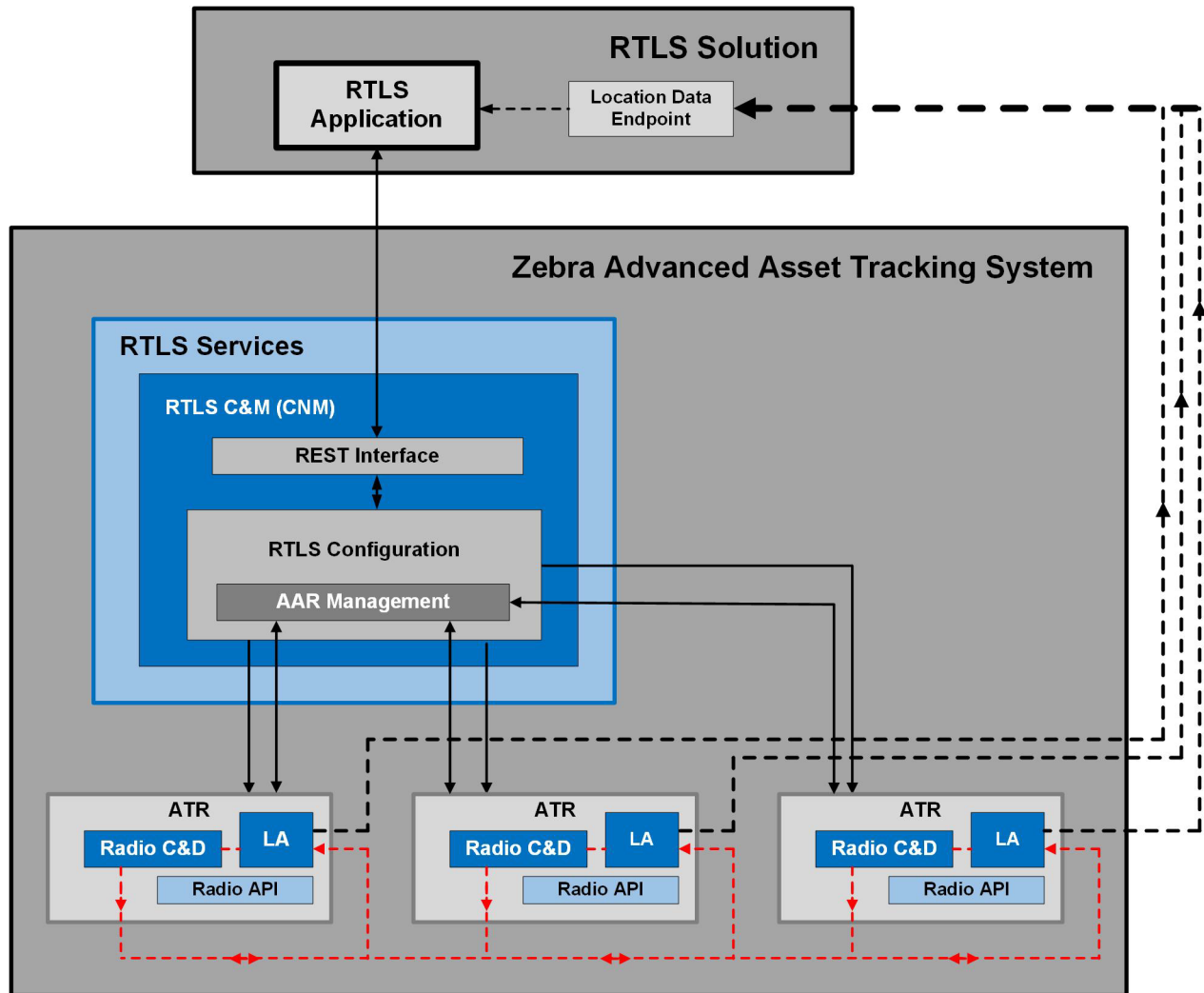


NOTE: Configuration and Location Analytics Software (CLAS) is needed to run RTLS Software. Configuration and Location Analytics Software (CLAS) is synonymous with RTLS Services software. The two terms are used interchangeably throughout this document.

Figure 1 illustrates the high level architecture of the ZAATS system showing the two main components of the ZAATS system:

- AARs (ATR7000 Advanced Readers)
- RTLS Services software.

Figure 1 Zebra Advanced Asset Tracking System (ZAATS)



In Figure 1, the solid lines correspond to configuration, control, and management interfaces. The dashed lines correspond to data interfaces. The dashed red lines carry tag ID bearing information that require high bandwidth and low-latency connections. Therefore, they typically reside on the same segment of a local area network.

ATR7000 Advanced Array Reader (AAR)

The ATR7000 Advanced Array Readers are EPC Gen2 readers with an integral phased array antenna capable of steering beams and estimating the bearing (angle of arrival) to EPC Gen2 tags. This product in the RFID portfolio is based on a Zebra proprietary advanced array architecture that provides unprecedented location accuracy and real time tracking of RFID tags.

A summary of the ATR7000 key product features are:

- Integral 14 element antenna array.
- Advanced multi-channel radio architecture provides accurate bearing estimations in a single read.
- Host platform compatible with Zebra's family of fixed RFID readers with support for embedded and external applications.
- Integration of Zebra's proprietary ASIC-based RFID radio.
- GPIO with external power for driving actuators and sensors.
- Support for several standard mounting options to simplify installation.
- Two power options: 802.3at Power over Ethernet (PoE+) or external +24 VDC power supply.
- Environmental specifications suitable for industrial and warehouse applications (-20°C to +55°C operation and IP51 sealing).

RTLS Services

RTLS Services (CLAS) serves as a data aggregator that executes location analytics to estimate the tag location and reports out unique tag ID, location, and time-stamp in real-time.

RTLS Services performs the following primary functions:

- Discovers readers on a local network.
- Configures each reader to read tags and report the estimated bearings.
- Estimates the tag location based on the bearings reported by the reader.
- Reports the location estimates to a location endpoint.
- Provides software interfaces to middleware applications that enable end-user solutions to associate items with identity, location and movement information, and delivers business logic to streamline operations or workflows.
- Provides an interface to end-users to configure and manage the RTLS system.
- Provides interface to manage and configure the ATR7000 readers in the facility.

The RTLS Services software consists of three major components:

- RTLS Configuration and Management Server (CNM)
- Location Analytics (LA)
- Radio Control & Data (CND).

RTLS Services is deployed as a group of Docker containers. A container is the mechanism that minimizes operating system and hardware dependencies, and isolates RTLS from the other software components that comprise a solution, allowing them to coexist on the same infrastructure. It is expected that RTLS Services typically reside on the same physical server as the solution.

A description of these and other components important to system operation are below.

RTLS Configuration and Management (CNM)

As shown in [Figure 1](#), RTLS Configuration and Management (CNM) is the primary component within RTLS Services responsible for managing and configuring the system, including system start and reset, reader discovery, initial and ongoing configuration of LA and CND, firmware and software upgrades, etc. CNM is the component of RTLS Services that resides on the server.

RTLS also provides the configuration and management interfaces (API) to the solution software through a RESTful interface, a common framework found in enterprise environments.

Location Analytics (LA)

[Figure 1](#) illustrates that Location Analytics (LA) is the primary component within RTLS Services responsible for aggregating bearing information received from the ATR7000 overhead readers, estimating x-y-z tag location, determining if a tag is moving (dynamic) or not moving (static), applying additional advanced algorithms that enhance static and dynamic location (tracking) accuracy, and reporting a final tag location estimate with metadata (EPC, timestamp, etc.) to a location data endpoint. LA also has the capability of combining raw bearing and location estimates from multiple RFID tags affixed to the same asset (for example, forklifts) to improve overall location accuracy and/or provide orientation and directionality information. The figure illustrates three AARs for simplicity, although, operation is designed to scale up to the maximum of 1000 readers per site. While LA is considered a component of RTLS Services, it is deployed by CNM to the readers at system start.

The interface between LA and the CND is optimized to be a high bandwidth, low latency one-way interface that carries only tag ID and bearing information, as indicated by the red arrows in the [Figure 1](#).

Radio Control & Data (CND)

The Radio Control & Data (CND) component is a reader-based application (process) that configures, controls, and maintains a connection to the RFID radio (engine), receives tag bearing reports from the radio and passes them to LA, and ensures the timestamps on the bearing reports are synchronized to the system time source.

While CND is considered a component of RTLS Services, it is deployed by CNM to the readers at system start.

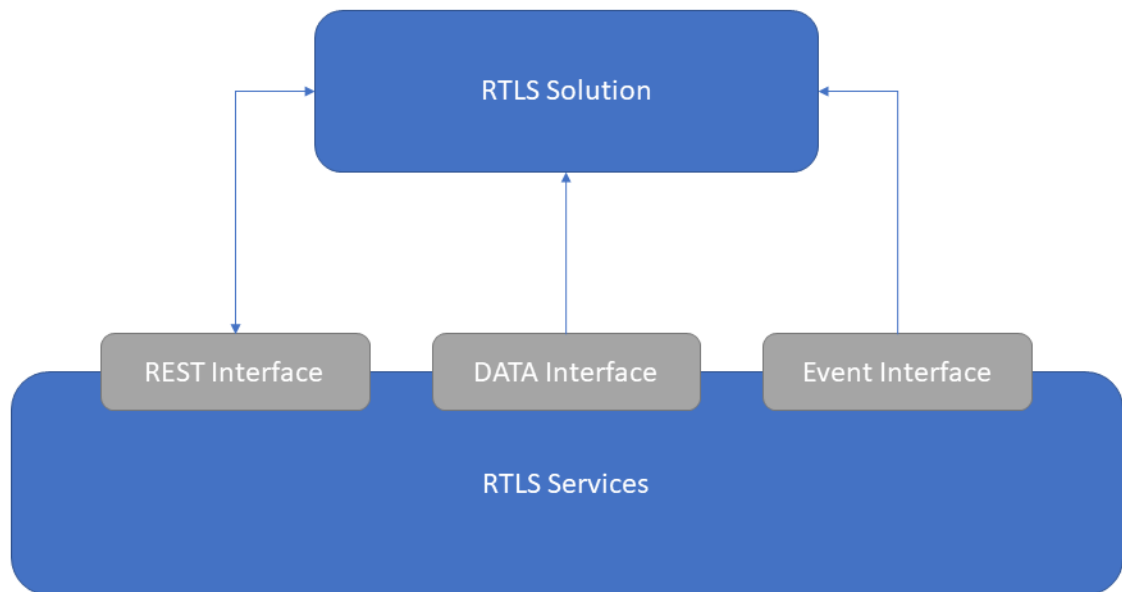
ZAATS Interfaces

ZAATS presents three main interfaces:

- A REST based management interface.
- A messaging stream interface for location data.
- A messaging stream interface for health and monitoring events.

The ZAATS REST API allows applications to view, configure, manage, and monitor various system components in the RTLS system. The ZAATS location data interface allows the client application to consume the location data output by the ZAATS system. The ZAATS event interface allows the client application to consume the health and monitoring events data output by the ZAATS system.

Figure 2 Interface Overview



REST Interface

The REST Interface is the primary mechanism to configure and manage the ZAATS system. It supports the ability to query the version, status, configuration of the RTLS system; start and stop the system; and reboot the ATR7000 readers. It also supports setting user-defined filters specifying the frequency and format of reported tag data.

Location Data Interface

Location update messages are sent from the LA components within RTLS through the Location Data Interface to a Location Data Endpoint (MQTT server or a Kafka broker). The RTLS customer's middleware application can consume these location update messages from the Location Data Endpoint to transform information about asset location into solutions that enhance efficiency and workflows of end user operations.

Events Interface

Health and Configuration events notification messages are sent from CNM within RTLS through the Events Interface to a Event Endpoint (MQTT server or a Kafka broker). The RTLS customer's middleware application can consume these event notification messages from the Event Endpoint to implement solutions for monitoring of the RTLS system and raise alerts to end users about system events.

Getting Started

Introduction

This chapter provides an overview of the steps involved in installing and setting up the CLAS and RTLS Services software, the primary software component of Zebra's Advanced Asset Tracking System (ZAATS). It also covers system requirements and prerequisites, including licensing, hardware, software, and networking.

Overview

The following details of an RTLS installation are covered as part of this guide.

- Setting up the system with prerequisites for running RTLS Services software.
- Installing and configuring RTLS Services software.
- Installing the RTLS simulator.
- Validating the RTLS server and software installation using the RTLS simulator.
- Running RTLS Services with ATR7000 hardware.

Hardware and other deployment aspects are covered in the ZAATS Deployment Guide. Refer to [Related Documents on page 7](#).

Hardware Requirements

The following are hardware requirements for a CLAS installation are as follows:

- Quad Core CPU @ 2.4 GHz
- 16 GB RAM
- 64 GB of free hard disk space.

Software Requirements

The operating system and software that must be installed on the server before installing CLAS and RTLS is as follows:

- Ubuntu v18.0.4 LTS or higher
- Docker v18.09 or higher

Refer to the following link to setup Docker:

<https://docs.docker.com/install/linux/docker-ce/ubuntu/>

Add the stable repository and not the nightly repository.

- Docker Compose v1.24.0 or higher

Refer to the following link to setup Docker Compose:

<https://docs.docker.com/compose/install/>

Install the stable release.

- ssh server (if remote access is required)

The latest version of openssh-server is recommended.

Registry Access

CLAS is deployed as a set of Docker images. To download the CLAS Docker images, access to the Zebra Docker Registry is required. The login credentials for Docker registry is sent to the customer via an email titled "CLAS (RTLS) SOFTWARE".

Installation Checklist

A CLAS and RTLS software installation involves several steps, all of which should be performed in a precise order. [Table 1](#) provides an outline of the steps. Each step is explained in detail in their respective sections.

Table 1 CLAS/RTLS Software Installation Steps

Step Number	Action Required	Section Reference	Comments
1	Perform prerequisite steps	Hardware Requirements on page 13 Software Requirements on page 14 Registry Access on page 14	None.
1a	Additional steps if installing RTLS behind a proxy	Installing RTLS Behind a Proxy Server on page 18	If RTLS is being installed behind a proxy, then RTLS and Docker must be configured to work behind a proxy.
2	Install RTLS	Installing CLAS Software on page 17	The next step is to install the RTLS Services software.
3	Configure RTLS	Configuring RTLS on page 29	This step involves editing the RTLS configuration files <code>rtls.conf</code> and <code>aar_info.csv</code> .
4	Validate RTLS installation	Validating a CLAS and RTLS Services Installation on page 37	This step involves installation of the necessary modules like the location endpoint and RTLS simulator to validate the RTLS installation.
5	Run RTLS with ATR7000s	Running CLAS with ATR7000 Hardware on page 41	This is the final step of an RTLS Services software installation. It is performed as part of the deployment process described in ZAATS Deployment Guide . Refer to Related Documents on page 7 .

[Table 2](#) includes the information required during installation and configuration. Make sure to have it available prior to beginning the installation.

Table 2 Required Installation and Configuration Items

Item	Comments
IP Address of Server	The IP address of the server that is reachable by the ATR7000s. The IP address is required for the <code>.env</code> file. See Table 3 on page 23 for more information.
Hostname/IP Address of location and events endpoint	The hostname and/or IP address and TCP Port of the location and events endpoint are required in the <code>rtls.conf</code> file. See Configuring RTLS on page 29 for more information.

Table 2 Required Installation and Configuration Items (Continued)

Item	Comments
TCP Ports	<p>RTLS uses the following set of ports on the host machine:</p> <ul style="list-style-type: none"> • 123 for NTP service • 20, 21 and the range of ports specified in RTLS_FTP_PASV_PORT_MIN and RTLS_FTP_PASV_PORT_MAX variables for FTP service • 5159, 5160, 5170, 5180, 5432 and the ports specified in REST_PORT while installing RTLS <p>See Table 3 on page 23 for more details on the configurable port ranges.</p>
Proxy Information	Required if installing RTLS behind a proxy. The configuration of the proxy is required.
Zebra Docker Registry Credentials	The CLAS Docker images are pulled from the Zebra Docker Registry in the Cloud. Therefore, access to the registry is a prerequisite for installation. Ensure that the user has Docker Registry credentials before starting the CLAS installation.

Installing CLAS Software

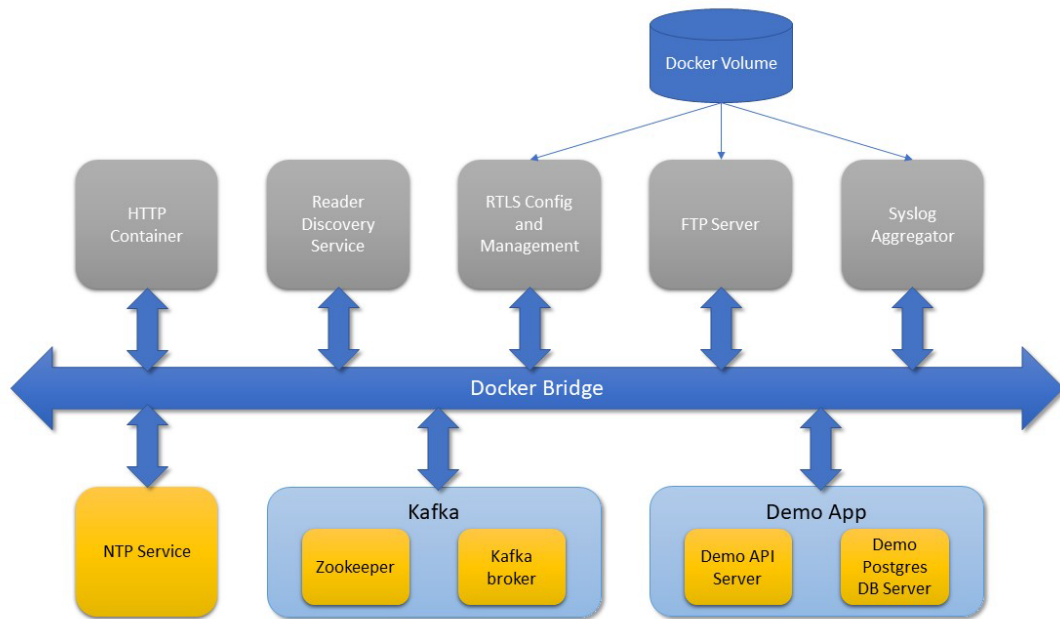
Introduction

This chapter describes steps required to install and configure CLAS/RTLS Services software. It explains different install time and runtime configurations and each configuration option in detail, including:

- Installing RTLS using the RTLS release package
- Installing certificates
- Changing the password for the REST Interface
- Configuring RTLS Services.

CLAS Docker Overview

An overview of the different Docker images is shown in [Figure 3](#). The services in grey are the base services. Regardless of the command line options, these services, and their associated Docker containers, are always started. Depending on the command line options, the optional services (in yellow) are started. For example, adding a -n switch to the `rtls start` command also starts an NTP service. This should be done only if the host running RTLS does not already have an NTP server running. For more information on different command line options, refer to [Starting RTLS on page 35](#).

Figure 3 Docker Images

Installing RTLS Behind a Proxy Server

If RTLS is being installed behind a proxy server, then Docker must be configured appropriately before RTLS is installed. If not installing RTLS behind a proxy, continue to [Installing RTLS on page 20](#).

As RTLS deployment involves the installation of various packages using the Ubuntu Package Manager, the HTTP/HTTPS proxy must be supplied. However, as RTLS Services also uses http to communicate with simulators and readers, the proxy must be disabled during runtime.

To setup the Docker service to use a proxy during deployment:

1. Create a system drop-in directory for the Docker service:
`sudo mkdir -p /etc/systemd/system/docker.service.d`
2. Create the appropriate configuration file.
 - a. For an HTTP proxy:
 - i. Create a file called `/etc/systemd/system/docker.service.d/http-proxy.conf` that adds the following HTTP_PROXY environment variable


```
[Service]
Environment = "HTTP_PROXY=http://proxy.example.com:80/"
```
 - b. For an HTTPS proxy:
 - i. Create a file called `/etc/systemd/system/docker.service.d/https-proxy.conf` that adds the following HTTPS_PROXY environment variable


```
[Service]
Environment = "HTTPS_PROXY=https://proxy.example.com:443/"
```

3. Flush changes:
`sudo systemctl daemon-reload`
4. Restart Docker:
`sudo systemctl restart docker`
5. Go to each of the following directories under the `rtls` directory and perform the steps below.
Directories: `cmm_client`, `disco`, `ftpd`, `httpd`, `loc_anal`, `rtls_configuration`
 - a. Create a file named `badproxy`. Copy and paste into the file the following:
`Acquire: : http: : Pipeline-Depth 0;`
`Acquire: : http: : No-Cache true;`
`Acquire: : BrokenProxy true;`
 - b. Edit the `Dockerfile` and copy and paste the following line after the first line `FROM ...`
`COPY ./badproxy /etc/apt/apt.conf.d/99fixbadproxy`

To setup Docker to remove proxy during runtime:

1. In the home directory of the user which starts the containers, create or edit the file `~/.docker/config.json`.
2. Add the following JSON example. Adding this JSON example clears the proxy settings at runtime.

```
{
  "proxies":
  {
    "default":
    {
      "httpProxy": ""
    }
  }
}
```

Installing RTLS

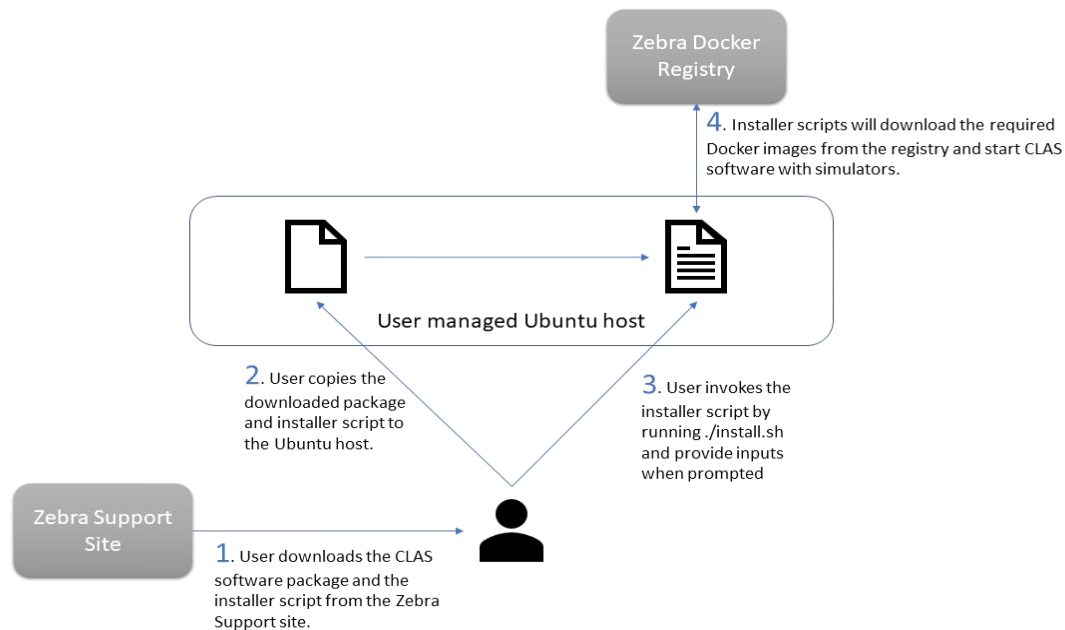
CLAS supports three methods of installation they are as follows:

1. Online installation with helper script: This method of installation is recommended if the target host on which the CLAS software is being installed has an Internet connection. The Internet connection is required to pull the CLAS docker images from the Zebra Docker registry. The helper script guides the user through the installation process by prompting user to enter various details required for installing CLAS software.
2. Online installation without helper script: This method is recommended for advanced users. This method also requires that the host on which the CLAS software is being installed has an Internet connection.
3. Offline installation: This method of installation allows users to install CLAS software on a host that does not have Internet access. However, users will require an intermediate host with Internet connection on which the offline package can be prepared for installation on the final host.

Online Installation with Helper Script

Installing RTLS services using this method requires that the user download the RTLS Services packages and the installer script from the Zebra Support site, run the installer script, and provide appropriate details when prompted.

Figure 4 CLAS Installation Process with Helper Script



Installation Procedure

To install RTLS:

1. Download the RTLS Services software package “rtls_services_cr_x.y.z.tar.gz” and the installer script “install.sh” from zebra.com/support.
2. Ensure that both the tar file and the installer script are in the same folder.
3. In a terminal, go to the directory where the files reside. For example, `cd rtls`.
4. To start the install, run the following command:
`./install.sh`

Installing CLAS Software



NOTE: Ensure that the installer script has executable permissions. Use the `chmod +x install.sh` command to assign executable permissions to this script.

- When the installer script begins, a confirmation appears asking to continue installing the script. Press `y`.
- If the host does not have Docker or Docker Compose installed, the script will first install Docker or Docker Compose, and then request the user to enter their credentials.

```
zebra@ubuntu:~/CLAS$ ls
install.sh  rtls_services_cr_2.2.43.99.tar.gz
zebra@ubuntu:~/CLAS$ ./install.sh
This script will install/upgrade the CLAS software on this host.
Continue? [y/n] y
Installing CLAS..
looking for docker-ce installation
dpkg-query: package 'docker-ce' is not installed and no information is available
Use dpkg --info (= dpkg-deb --info) to examine archive files,
and dpkg --contents (= dpkg-deb --contents) to list their contents.
Docker not installed.Installing docker..
Do you want to install docker? [y/n] y
[sudo] password for zebra: █
```

- Once Docker and Docker Compose are installed, the script prompts the user to provide details to configure the CLAS installation.

```
Checking for existing CLAS installations..
CLAS will be installed in /opt/rtls. Hit Enter to continue or specify a new path:
Creating directory for new installation..
*****
IMPORTANT: CLAS needs to know the details of the network interface on this host.
Please make sure that the selected network interface is on the same subnet as the ATR7000s.
*****
Found the following network interface: [ens33 : 192.168.1.39]
Continue using this interface? [y/n]: y
configuring CLAS installation.
CLAS has a bundled Kafka broker and will be configured to use that by default.
If you want to change this behaviour, select 'n' else select 'y'.
Use bundled Kafka broker? [y/n]: y
Logging into docker registry. Please Enter your username and password.
The registry credentials are send to you in a email titled 'CLAS Beta' or 'CLAS software'.
Username: █
```

- Installation Path: To install in the default path hit “Enter” when prompted or enter a new path.
 - Network interface: The installer script will detect the network interfaces and prompt the user for confirmation on which network interface to use. If there are multiple network interfaces on the host, then enter the name of the network interface to be used for configuring CLAS software.
 - Kafka Broker: CLAS software comes bundled with Kafka broker. If the user wants to use the bundled Kafka broker, then press “y”. If any Kafka broker other than the bundled broker must be used then press “n,” and then enter the IP:PORT of the Kafka broker.
 - Docker registry credentials: When the installer is ready to download the CLAS docker images, the system will prompt the user for their Zebra Docker registry credentials. Enter credentials when prompted. The credentials are provided to the user via an email from Zebra.
- Once the Docker login is successful, the installer script prompts the user for confirmation to download the docker images and start CLAS in the simulator mode.

```
Password:
WARNING! Your password will be stored unencrypted in /home/zebra/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
Checking if NTP service is installed..
NTP service is not installed.
adding bundled kafka option.
Start CLAS in sim mode? This will pull all the docker images from the registry and start CLAS. [y/n]
```

9. Downloading the images and starting CLAS will take some time. Once the download is complete and CLAS is started in simulator mode, the installer will exit, and the installation is complete.

```
Successfully installed CLAS. Please go to the following URL in your browser.

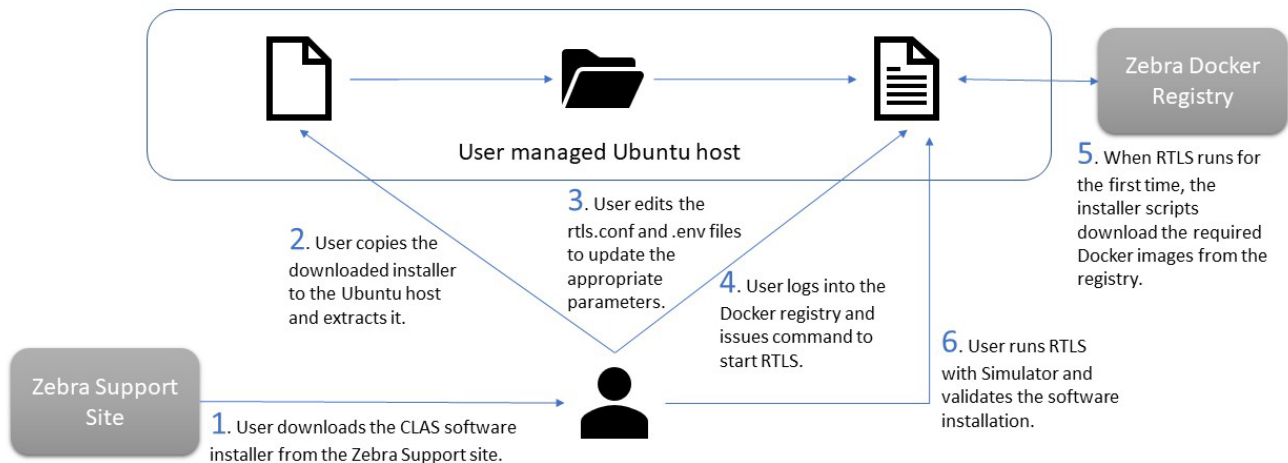
https://192.168.1.39:80
To stop CLAS run 'cd /opt/rtls/2.2.43.99/rtls && sudo ./rtls.sh -sdnk stop'
To start CLAS any time run 'cd /opt/rtls/2.2.43.99/rtls && sudo ./rtls.sh -dnk start'
zebra@ubuntu:~/CLAS$
```

Online Installation without Helper Script

Installing RTLS Services requires downloading the RTLS Services package from the Zebra Support site, extracting the package, modifying the appropriate parameters, logging into the Zebra Docker Registry, and using the extracted scripts to get the Docker images from the Zebra Docker Registry.

The CLAS software installation process is as shown in Figure 5.

Figure 5 CLAS Installation Process



To begin the installation:

1. Download the CLAS software package from the Zebra Support site and extract it on the Ubuntu host where CLAS will be installed.
2. Edit the `.env` and `rtls.conf` files to provide appropriate parameters.
3. After the configuration files are setup, perform a Docker login using the credentials provided in the email from Zebra titled “CLAS (RTLS) Software”, and then run the `rtls.sh` script to start RTLS. Upon starting RTLS for the first time, the RTLS installer scripts download the Docker images from the Zebra Docker Registry and starts RTLS.

4. To validate the RTLS server setup, perform a setup validation using the RTLS Simulators.

Depending on the command line options, the appropriate Docker images are pulled from the Docker Registry when the `rtls.sh` script is run.

Installation Procedure

To install RTLS:

1. Download the RTLS Services software package (`rtls_services_cr_x.y.z.tar.gz`) from www.zebra.com/support.
2. Extract the RTLS release package using the following command:

```
tar -xzf rtls_services_cr_x.y.z.tar.gz
```
3. Change to the `rtls` directory using the following command:

```
cd rtls
```
4. Edit the `.env` file and set the parameters accordingly. The descriptions for the parameters in this file are provided in [Table 3 on page 23](#).



NOTE: Parameters are case sensitive.

Under typical circumstances, only the `RTLS_AAR_IFC_IP` in the `.env` file require to be updated.

5. Log into the Zebra Docker Registry using following command:

```
docker login registry.devsecops.zebra.com -u <username> -p <password>
```



NOTE: The user will receive credentials in an email from Zebra titled “CLAS (RTLS) Software”.

6. Start RTLS using the following command:

```
./rtls.sh start
```



NOTE: The user running `rtls.sh` must be in the Docker group. If the user is not, `sudo` should be used when running `rtls.sh`.

Table 3 Parameter Descriptions of the `.env` file

Parameter Name	Description	Default
REST_PORT	This parameter configures the port on which the RTLS server exposes its REST interface.	80
RTLS_AAR_IFC_IP	<p>Setting this parameter is mandatory.</p> <p>If the RTLS host machine has a single network interface, this parameter should be set to the IP address of the host machine that is running on the RTLS server.</p> <p>In cases where the host machine has multiple network interfaces, this parameter should be set to the IP of the network interface that is on the same subnet as the ATRs.</p>	0.0.0.0

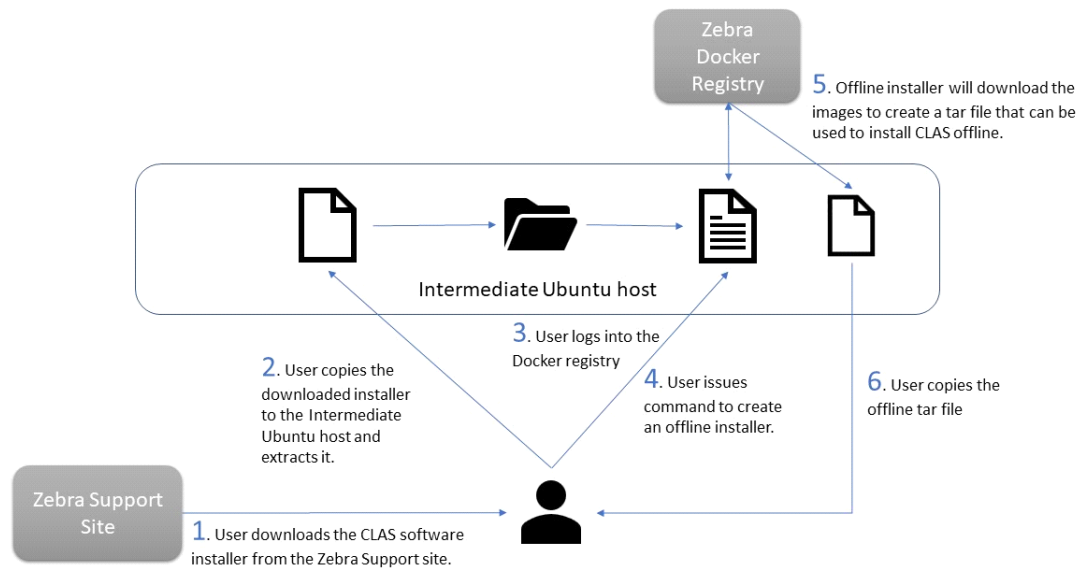
Parameter Name	Description	Default
UPSTREAM_NTP_ADDR	RTLS Services includes an NTP server that is used by the ATR7000 to synchronize time. The bundled NTP service is started only when RTLS is started using the -n switch. The NTP server on the RTLS Services needs an upstream NTP server with which to synchronize. The value of this parameter should be set to the corporate NTP server addresses.	time1.google.com
RTLS_FTP_PASV_PORT_MIN	The RTLS Services package also includes an FTP/FTPS server. As the RTLS FTP server runs in passive mode, it requests a range of ports to be assigned to it. This parameter must contain the starting point in this range.	50000
RTLS_FTP_PASV_PORT_MAX	This parameter must be assigned the end port in the range of ports assigned for FTP passive mode operation.	50200
CERTIFICATES_PATH	This parameter must be set to the path where the certificates that need to be installed are located. The default shared_vol indicates to RTLS to create its own self-signed certificates. For more information, see Installing Certificates .	shared_vol
SSL_ENABLE	This parameter must be set to YES or NO. If set to YES, then RTLS is started with the REST interface in HTTPS mode. If set to NO, then RTLS is started with the REST interface in HTTP mode.	YES
RSYSLOG_PORT	The port number on the Ubuntu host that will be used for reader syslog aggregator service. This parameter can be configured to any of the available ports on the Ubuntu host.	40000
DEMO_DB_PORT	This is only applicable if the user is running the CLAS demo application, otherwise it will be ignored. This variable configures the port number, which the demo application database server binds to.	5432
DOCKER_REGISTRY_ADDRESS	This variable configures the Zebra Docker Registry that must be used to pull the CLAS Docker images. Users never have to change this unless explicit required.	registry.devsecops.zebra.com

Offline Installation

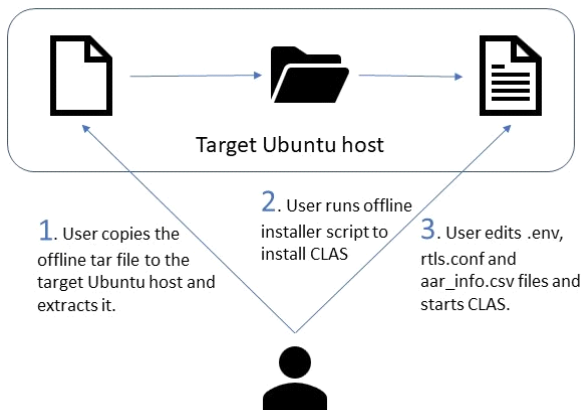
Use this method to install CLAS if it must be installed on a host that does not have an Internet connection. Installing CLAS using this method is a two-step process.

1. Creating an offline installer package: The user downloads the CLAS software package from Zebra Support site on to an intermediate host that has Internet access, and then uses the offline installer script embedded in the CLAS packages to create a tar file that has the CLAS Docker images.

Installing CLAS Software



2. Installing CLAS using the offline installer package: The user uses the offline packages created in the previous step and installs it on the target host.



Installation Procedure

To create an offline installer package on an Ubuntu host that has Internet connection:

1. Download the RTLS Services software package (`rtls_services_cr_x.y.z.tar.gz`) from www.zebra.com/support.
2. Run the following command to extract the RTLS release package:
`tar -xzf rtls_services_cr_x.y.z.tar.gz`
3. Change to the `rtls` directory using the following command:
`cd rtls`
4. Run the following command to log into the Zebra Docker Registry:
`docker login registry.devsecops.zebra.com -u <username> -p <password>`



NOTE:Users will receive credentials in an email from Zebra with the subject line “CLAS (RTLS) Software.”

5. Run the following command to create an offline installer package:
`./rtls_offline_installer.sh create`



NOTE:Users running `rtls.sh` must belong to the Docker group. If a user is not in the Docker group, then use `sudo` when running `rtls.sh`.

6. After the script finishes running, run the `cd` command to go to the previous folder.
7. Copy the offline package (`clas_offline_package.tar.gz`) to the target host.

To install CLAS software from an offline package:

1. Copy `clas_offline_package.tar.gz` to the host where CLAS is to be installed.
2. Extract the offline installer package `tar -xzf clas_offline_package.tar.gz`.
3. Type the following command to go to the `rtls` folder:
`cd rtls`
4. Run the following command to install CLAS:
`./rtls_offline_installer.sh install`
All docker images are loaded on the host.
5. Edit the `.env` file and set the parameters as required. The descriptions for the parameters in this file are provided in [Table 3 on page 23](#).



NOTE:Parameters are case sensitive. Typically only the `RTLS_AAR_IFC_IP` in the `.env` file requires to be updated.

6. Run the following command to start RTLS:
`./rtls.sh start`



NOTE:Users running `rtls.sh` must belong to the Docker group. If a user is not in the Docker group, then use `sudo` when running `rtls.sh`.

Installing Certificates

There are three sets of certificates that can be installed in RTLS, all of which must be present in the path mentioned in the `CERTIFICATES_PATH` parameter with different file names as described later. The certificates are:

- Certificates for secure REST interface
- Certificate for secure Kafka communication
- Certificate for secure MQTT communication.

Installing Certificates for Secure REST Interface

When supplying certificates for the REST interface, the following conditions must be met for certificates to be installed properly:

- The REST interface certificate and key must be supplied in pem format.
- The certificate and key must have the file name `rtls_server.crt.pem` and `rtls_server_key.pem` respectively.
- The key must not be password protected.

If there is no user-supplied certificates and `SSL_ENABLE` is set to `YES`, the `CERTIFICATE_PATH` should remain the default (`shared_vol`). This causes RTLS to create self-signed certificates for use on the REST interface.

Installing Certificates for Secure Kafka Communication

RTLS supports publishing location and health, and monitoring to a secure Kafka broker. If the Kafka broker is configured for SSL communication, the client certificate must be supplied while deploying RTLS. The client certificate must be present in the `CERTIFICATES_PATH` as mentioned in `.env` and it must be named `ca-cert`.

When RTLS starts, it looks for a certificate that is named `ca-cert` in `CERTIFICATES_PATH` and uses it for securing Kafka communication. If a certificate with the name `ca-cert` is not found at startup, RTLS uses a non-secure plain-text mode of communication with the Kafka broker.

Installing Certificates for Secure MQTT Communication

RTLS supports publishing location and health, and monitoring to a secure MQTT broker. If the MQTT broker is secured using certificates, then the client certificate must be supplied while starting RTLS. The client certificate must be present in the `CERTIFICATES_PATH` as mentioned in the `.env` file, and it must be named `mqtt-ca-cert`. When RTLS starts, it looks for a certificate that is named `mqtt-ca-cert` in `CERTIFICATES_PATH`, and uses it for securing MQTT communications. If a certificate with the name `mqtt-ca-cert` is not found at startup, RTLS uses a non-secure plain-text mode of communication with the MQTT broker.

Changing the Password for the REST Interface

By default, the REST interface uses the following credentials:

- Username: `rtlsadmin`
- Password: `Z@@t$R1I$`

These credentials can be changed after deploying RTLS Services. Changing the user password is only possible while RTLS is running.

To change the user password:

1. Enter the `rtls` directory using the following command:

```
cd rtls
```

2. Start `rtls` using the following command:

```
./rtls.sh start
```



NOTE: The user running `rtls.sh` must be in the Docker group. If the user is not, `sudo` should be used when running `rtls.sh`.

3. Run the following command:

```
docker exec -it rtls-httpd-container bash
```



NOTE: The user running Docker must be in the Docker group. If the user is not, `sudo` should be used when running Docker.

4. Once inside the container, run the following commands:

```
/usr/local/apache2/bin/htpasswd -b
```

```
/usr/local/apache2/conf/password rtlsadmin <NEW_PASSWORD>
```

Replace the new password with the desired password. Once changed, the password changes persist even after restarting RTLS.

Configuring RTLS

RTLS Services is configured through two configuration files, `rtls.conf` and `aar_info.csv`. The `rtls.conf` file is a text file that controls the runtime configuration of RTLS and includes both user changeable and auto filled parameters. The `aar_info.csv` file is described in [Adding ATRs to RTLS on page 34](#).

The default `rtls.conf` file looks like the file shown below. The parameters in the private section of `rtls.conf` are automatically edited by the RTLS deploy scripts and should not be edited by the user. However, the other values can be edited.

[Table 4](#) includes the parameters available in `rtls.conf` for basic configuration of RTLS services. [Table 5](#) includes additional parameters available in `rtls.conf` for advanced configuration of RTLS services.

Default contents `rtls.conf` file

```
[rtls]
# This section consists of user changeable parameters.
#Type of the location data_endpoint. Only kafka/mqtt are supported
location_endpoint_type = kafka
# Address of the location data_endpoint
location_endpoint_addr = 0.0.0.0:9092
# Name of the topic of the data_endpoint
location_endpoint_topic = rtls.tag_location_update.v2.json
#Type of the events endpoint. Only kafka/mqtt are supported.
events_endpoint_type = mqtt
#Address of the events endpoint.
events_endpoint_addr = 0.0.0.0:1883
#Name of the topic for events
events_topic = rtls.events.json
#Site id for LA
location_analytics_site_id = 1
#Global Error threshold value for reference tag event generation
reference_tag_error_threshold = 8
#A rolling window in minutes over which the refernce tag statistics are calculated
reference_tag_window = 5
#Enables or disables ATR power negotiation when powered via POE+ switches. can be set
to enable or disable.
lldp = disable
[location_analytics]
# set this to auto to enable auto start of LA and manual to disable autostart of LA
location_analytics_start = auto
#This will set the units of distance in LA to feet or meters
location_analytics_config_units = feet
#This will set the reporting fields in LA
```

```

location_analytics_reporting_fields =
report_source,confidence,position,timestamp,epc_id
#This will set the time threshold after which the tag report will be published
la_time_filter = 1
#This will set the confidence threshold above which the tag report will be published
la_confidence_filter = 70
#This will set the velocity threshold in ft/sec or mts/sec depending on the units
configured in
location_analytics_config_units above which the tag report will be published
la_velocity_filter = 1000
la_id_filter = 00000000000000000000000000000000
la_id_filter_mask = 00000000000000000000000000000000
la_id_filter_num_bytes = 12
la_static_or_dynamic_filter = ignore
la_distance_filter = 5
la_fixed_z = 3.0
la_static_v_dynamic_threshold = 2.0
[radio_c_and_d]
# set this to auto to enable autostart of CND and manual to disable auto start of CND
radio_c_and_d_start = auto
# set this to bearing to get real tag bearings or sim to get run simulation
radio_c_and_d_config = bearing
# the amount by which the accelerometer roll or pitch must change for an event to be
sent
accelerometer_event_threshold = 1.0
[private]
#This will change the logging level for rtls config. string should be one of DEBUG,
INFO, WARN, ERROR
log_level = INFO
location_analytics_address = 0.0.0.0
location_analytics_version = 1.0.0.7
radio_c_and_d_app_version = 1.0.2.6
docker_host_address = 0.0.0.0
[deprecated]
kafka_brokers = 0.0.0.0:9092

```

Basic Configuration Options

Table 4 Basic RTLS Configuration Options

Parameter Name	Description	Allowed Options	Default
location_endpoint_type	The type of message broker over which the RTLS publishes the location estimates.	kafka, mqtt	kafka
location_endpoint_topic	The topic on the message broker as configured in the location_endpoint_type over which the RTLS publishes the location estimates.	A valid string naming the topic for the configured broker.	rtls.tag_location_update.v2.json
location_endpoint_addr	The IP:PORT or HOSTNAME:PORT combination of the broker to which RTLS publishes location estimates. Important: Changing this value is mandatory.	A valid IP:PORT combination or a HOSTNAME:PORT combo that is reachable from LAs. This address must be accessible from the ATRs.	0.0.0.0:9092
reference_tag_error_threshold	Global reference tag error threshold value for event generation.	A valid integer specifying the error threshold in units of distance as configured in the location_analytics_config_units parameter.	8
reference_tag_window	A rolling window in minutes over which the reference tag statistics are calculated.	A valid integer specifying the time window in minutes.	5
events_endpoint_type	The type of message broker over which RTLS publishes warning and error events that occur in the system.	kafka, mqtt	mqtt
events_endpoint_addr	The IP:PORT or HOSTNAME:PORT combination of the broker to which RTLS publishes system events.	A valid IP:PORT combination or a HOSTNAME:PORT combination that is reachable from the RTLS server.	0.0.0.0:1883

Table 4 Basic RTLS Configuration Options (Continued)

Parameter Name	Description	Allowed Options	Default
events_topic	The topic on the message broker as configured in the events_endpoint_type over which RTLS publishes system events.	A valid string naming the topic for the configured broker.	rtls.events.json
lldp	This option decides whether the CNM will configure LLDP on the readers at startup. If set to enable, CNM will configure the readers to perform LLDP power negotiation on startup. If set to disable, CNM will not take any action on the ATR7000 power negotiation at startup.	enable, disable	disable

Advanced Configuration Options

Table 5 Advanced RTLS Configuration Options

Parameter Name	Description	Allowed Options	Default
location_analytics_start	This option changes the default startup behavior of Location Analytics. If set to manual, RTLS does not start and initialize the LA at startup. The LAs must be initialized using the REST APIs. This is only required to be set to manual for debugging purposes, the user should never have to change the setting to anything other than auto.	auto, manual	auto
location_analytics_config_units	The units of distance that should be used in LA location estimates.	feet, meters	feet
location_analytics_reporting_fields	A comma separated string that specifies what fields must be included in the location estimate report to location endpoint.	message_id, report_source, confidence, position, timestamp, epc_id, readers, velocity, direction, static_or_dynamic	report_source, confidence, position, timestamp, epc_id
la_time_filter	This option sets the time threshold after which a tag location must be reported to location endpoint.	floating point number representing time in seconds	1.0

Table 5 Advanced RTLS Configuration Options (Continued)

Parameter Name	Description	Allowed Options	Default
la_confidence_filter	This option sets the minimum confidence a location estimate must be reported to location endpoint, even if the tag has not moved by the amount configured in the distance filter.	integer in percentage	70
la_velocity_filter	This option sets the velocity threshold in ft/sec or m/sec, depending on the units configured in location_analytics_config_units parameter, above which a location report is sent to location endpoint.	integer value representing distance/sec	1000
la_id_filter	This option sets the filter that should be applied to epc_id of the tags. All tags matching this tag pattern is reported to the location endpoint.	hex string	0000000 0000000 0000000 000
la_id_filter_mask	The mask to be used to apply the tag ID filter.	hex string	0000000 0000000 0000000 000
la_id_filter_num_bytes	The length of the tag ID for tag ID filter.	integer representing the number of bytes in the tag ID	12
la_static_or_dynamic_filter	Filter that specifies whether to publish tag location reports for only static or only dynamic or both.	static_only, dynamic_only, ignore	ignore
la_distance_filter	This option specifies the distance a tag needs to move for it to be reported to location endpoint. The units of distance used is the one specified in location_analytics_config_units parameter.	floating point number	5.0
la_fixed_z	The fixed_z value that is used for estimating locations.	floating point value	3.0
la_static_v_dynamic_threshold	The threshold of movement of the tag which is used to determine if the tag is static or dynamic.	floating point value	2.0
radio_c_and_d_config	This option starts RTLS bearing.	bearing	bearing

Table 5 Advanced RTLS Configuration Options (Continued)

Parameter Name	Description	Allowed Options	Default
accelerometer_event_threshold	The amount by which the accelerometer roll or pitch must change for an event to be sent.	A number indicating the threshold by which the roll and pitch change in ATR generates an event.	1.0
radio_c_and_d_start	This option changes the default startup behavior of CND. If set to manual, RTLS does not start and initialize the CNDs at startup. The CNDs have to be initialized using the REST APIs. This is only required to be set to manual for debugging purposes, the user should never have to change the setting to anything other than auto.	auto, manual	auto

Adding ATRs to RTLS

In an actual RTLS deployment, the aar_info.csv file must contain reader information about the physical system.



NOTE: The process of defining the ATR host names, IP addresses, and determining x-y-z coordinates, and orientation is described in the ZAATS Deployment Guide. Refer to [Related Documents on page 7](#).

The user is required to enter the following reader information in CSV format, in this order:

1. ATR Host Name
2. IP Address
3. x coordinate of the reader
4. y coordinate of the reader
5. z coordinate of the reader
6. Orientation of the reader

The contents of a sample aar_info file is as follows:

```
AAR Host Name, IP Address, x, y, z, Orientation
ATR7000F422C8,192.168.7.201,40.0,100,17.1,0
ATR7000F476E1 192.168.7.202,15.0,87.5,17.1,-5
ATR7000F3F489 192.168.7.203,40.0,75.0,17.1,0
ATR7000F3F316 192.168.7.204,15.0,62.5,17.1,0
ATR7000F3F4A1 192.168.7.205,40.0,50.0,17.1,10
```



NOTE: The units of distance used to supply the coordinates of ATRs should be on the same as mentioned in rtls.conf file in the location_analytics_config_units field.

Preparing to Run RTLS

Before starting RTLS:

- Stop any FTP service on the host.
- Ensure the port ranges mentioned in the .env file for FTP (RTLS_FTP_PASV_PORT_MIN and RTLS_FTP_PASV_PORT_MAX) are available and are not being used by any other application.
- Ensure the port assigned to RTLS for REST Service in the .env file, 80 by default, is available.

Starting and Stopping RTLS

Starting RTLS

After installing and configuring RTLS software and adding readers, the system is ready to start.

To start RTLS:

1. Enter the rtls directory using the following command:

```
cd rtls
```

2. Log into the Zebra Docker Registry using following command:

```
docker login registry.devsecops.zebra.com -u <username> -p <password>
```



NOTE: The RTLS Docker images are pulled from the registry only when they are started for the first time. Although a Docker login step is not generally required, it may be needed if a new add-on service that is being started for the first time. A list of all add-on services is mentioned in the next step.

3. Start rtls using the following command:

```
./rtls.sh start
```

The above command starts a base set of RTLS services to run. This command satisfies most use cases. However, RTLS provides other add-on services that must be started depending on the RTLS configuration in rtls.conf file or the host server setup. The usage syntax of rtls.sh script are as follows:

```
./rtls.sh [options] start
```

- options:
 - -n: start the NTP service. Use this option only if the host server is not already running an NTP service. If this option is supplied, then make sure the NTP service is stopped on the host server.
 - -d: start the Demo application. Using this option starts the Demo application along with RTLS.
 - -k: RTLS comes with a bundled Kafka broker. Using this option starts the bundled Kafka broker along with RTLS. This is not required if a third party Kafka broker is being used.
 - -s: start RTLS in simulation mode along with the simulators.
- Example Usage:
 - ./rtls.sh -n start: start RTLS with the bundled NTP service.
 - ./rtls.sh -dk start: starts RTLS with the demo application and the bundled Kafka broker.
 - ./rtls.sh -nsdk start: starts RTLS with the bundled NTP server, demo application, and the bundled Kafka broker in simulation mode. This will also start the default simulation containers.



NOTE: The user running `rtls.sh` must be in the Docker group. If the user is not in the Docker group, use `sudo` when running `rtls.sh`.

This starts RTLS, which in turn connects to all the specified readers and starts reading and locating tags. Once RTLS Services are started, tag location estimates are published to the configured broker serving as the location endpoint. The end user application can run a consumer and consume the location data from the broker to verify the RTLS installation.



NOTE: If the Ubuntu host is not already running an NTP server, then add the `-n` option to the `rtls.sh` command.

Stopping RTLS

To stop RTLS, run the following commands:

```
cd rtls
./rtls.sh -nsdk stop
```



NOTE: The user running `rtls.sh` must be in the Docker group. If the user is not in the Docker group, use `sudo` when running `rtls.sh`.

The above command stops RTLS and stops the flow of location estimates to the broker.

Validating a CLAS and RTLS Services Installation

Introduction

The ZAATS Deployment Guide (refer to [Related Documents on page 7](#)) describes in detail the steps in an RTLS installation, although it does not cover software installation, which is typically deployed independent of the on-site hardware. This document focuses on the steps involved in deploying RTLS Services software. This chapter focuses on how to validate the RTLS Services installation. The following sections provide information about the RTLS Simulator architecture and how to setup external components like the Kafka broker, and how to configure RTLS to work with Simulators.

RTLS Simulator Architecture

The primary function of the RTLS Simulator is to enable testing of RTLS Services in the absence of physical ATR7000 readers. The RTLS Simulator is an application that emulates ATR7000 readers and their associated tag and bearing reports, and the LA component of RTLS Services publishes tag data reports to the location endpoint.

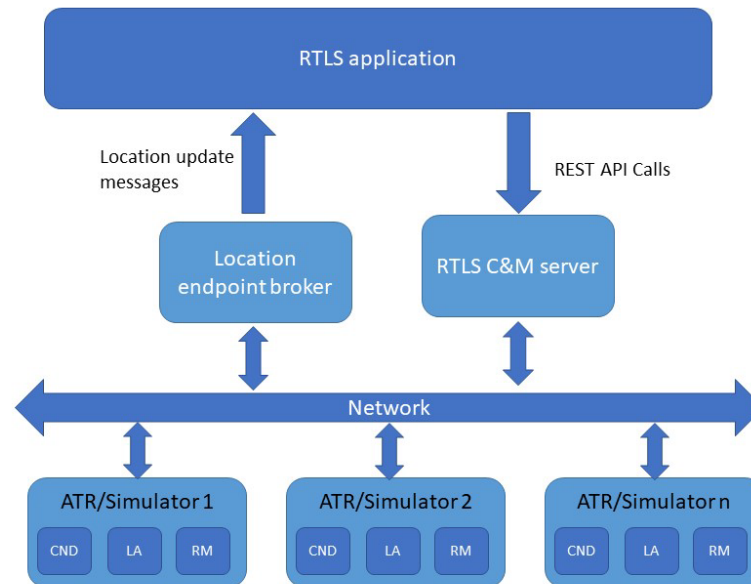
The functionalities provided by the RTLS Simulator are:

- ATR RM interface emulator provides reader management functionality (required for seamless interfacing to the RTLS C&M server).
- CND emulator provides tag bearing reports.
- LA functionality provides location estimates.

By default, the RTLS Simulator simulates 28 readers and generates tags at the rate of 120 tag reports per second.

[Figure 6](#) illustrates the architecture of RTLS when using a RTLS Simulator. The blocks that are named ATR/Simulator 1 to ATR/Simulator n are all simulated by the Simulator.

Figure 6 RTLS Architecture and Setup in Simulation Mode



Starting RTLS in Simulation Mode

To start RTLS in a simulation mode:

1. Change to the `rtls` directory and login to the Zebra Docker Registry using the following commands:

```
cd rtls
```

```
docker login registry.devsecops.zebra.com -u username -p password
```
2. Start RTLS in simulation mode along with the bundled Kafka broker using the following command:

```
./rtls.sh -sk start
```



NOTE: The user running `rtls.sh` must be in the Docker group. If the user is not in the Docker group, use `sudo` when running `rtls.sh`.

Verification

To verify that RTLS is reporting location estimates on the Kafka broker, run the following command:

```
docker exec -it rtl_s_bi tname_i_kafka_container kafka-console-consumer.sh --bootstrap-server
<kafka_broker_ip>:<port> --topic rtl_s_tag_location_update.v2.json
```

The kafka_broker_ip and port in the above command must be same as what is supplied in rtl_s.conf file.

If the system is properly configured, a steady stream of messages must be seen in the Kafka consumer console as indicated in [Figure 7](#).

Figure 7 Kafka Consumer

```
rpatil@rpatil-HP-Z440-Workstation: ~/kafka/kafka-docker
{"message_id":1445,"epc_id":"3175050952000030de000002","timestamp":"2018-07-20T10:47:01.239+0000","position":{"x": 9.42,"y": 15.75,"z": 3.00},
"confidence":50.00}
{"message_id":1446,"epc_id":"3175050952000030de000000","timestamp":"2018-07-20T10:47:05.104+0000","position":{"x": 12.24,"y": 13.65,"z": 3.00},
"confidence":50.00}
{"message_id":1447,"epc_id":"3175050952000030de000001","timestamp":"2018-07-20T10:47:05.108+0000","position":{"x": 10.12,"y": 10.82,"z": 3.00},
"confidence":50.00}
{"message_id":1448,"epc_id":"3175050952000030de000002","timestamp":"2018-07-20T10:47:06.141+0000","position":{"x": 9.42,"y": 15.75,"z": 3.00},
"confidence":50.00}
{"message_id":1449,"epc_id":"3175050952000030de000000","timestamp":"2018-07-20T10:47:10.008+0000","position":{"x": 12.24,"y": 13.65,"z": 3.00},
"confidence":50.00}
{"message_id":1450,"epc_id":"3175050952000030de000001","timestamp":"2018-07-20T10:47:10.008+0000","position":{"x": 10.12,"y": 10.82,"z": 3.00},
"confidence":50.00}
{"message_id":1451,"epc_id":"3175050952000030de000002","timestamp":"2018-07-20T10:47:11.040+0000","position":{"x": 9.42,"y": 15.75,"z": 3.00},
"confidence":50.00}
{"message_id":1452,"epc_id":"3175050952000030de000001","timestamp":"2018-07-20T10:47:15.160+0000","position":{"x": 10.12,"y": 10.82,"z": 3.00},
"confidence":50.00}
{"message_id":1453,"epc_id":"3175050952000030de000000","timestamp":"2018-07-20T10:47:15.161+0000","position":{"x": 12.24,"y": 13.65,"z": 3.00},
"confidence":50.00}
{"message_id":1454,"epc_id":"3175050952000030de000002","timestamp":"2018-07-20T10:47:16.192+0000","position":{"x": 9.42,"y": 15.75,"z": 3.00},
"confidence":50.00}
{"message_id":1455,"epc_id":"3175050952000030de000000","timestamp":"2018-07-20T10:47:20.059+0000","position":{"x": 12.24,"y": 13.65,"z": 3.00},
"confidence":50.00}
{"message_id":1456,"epc_id":"3175050952000030de000001","timestamp":"2018-07-20T10:47:20.062+0000","position":{"x": 10.12,"y": 10.82,"z": 3.00},
"confidence":50.00}
{"message_id":1457,"epc_id":"3175050952000030de000002","timestamp":"2018-07-20T10:47:21.095+0000","position":{"x": 9.42,"y": 15.75,"z": 3.00},
"confidence":50.00}
{"message_id":1458,"epc_id":"3175050952000030de000000","timestamp":"2018-07-20T10:47:25.216+0000","position":{"x": 12.24,"y": 13.65,"z": 3.00},
"confidence":50.00}
{"message_id":1459,"epc_id":"3175050952000030de000001","timestamp":"2018-07-20T10:47:25.225+0000","position":{"x": 10.12,"y": 10.82,"z": 3.00},
"confidence":50.00}
{"message_id":1460,"epc_id":"3175050952000030de000002","timestamp":"2018-07-20T10:47:26.002+0000","position":{"x": 9.42,"y": 15.75,"z": 3.00},
"confidence":50.00}
{"message_id":1461,"epc_id":"3175050952000030de000001","timestamp":"2018-07-20T10:47:30.115+0000","position":{"x": 10.12,"y": 10.82,"z": 3.00},
"confidence":50.00}
{"message_id":1462,"epc_id":"3175050952000030de000000","timestamp":"2018-07-20T10:47:30.115+0000","position":{"x": 12.24,"y": 13.65,"z": 3.00},
"confidence":50.00}
{"message_id":1463,"epc_id":"3175050952000030de000002","timestamp":"2018-07-20T10:47:31.143+0000","position":{"x": 9.42,"y": 15.75,"z": 3.00},
"confidence":50.00}
{"message_id":1464,"epc_id":"3175050952000030de000000","timestamp":"2018-07-20T10:47:35.015+0000","position":{"x": 12.24,"y": 13.65,"z": 3.00},
"confidence":50.00}
{"message_id":1465,"epc_id":"3175050952000030de000001","timestamp":"2018-07-20T10:47:35.016+0000","position":{"x": 10.12,"y": 10.82,"z": 3.00},
"confidence":50.00}
{"message_id":1466,"epc_id":"3175050952000030de000002","timestamp":"2018-07-20T10:47:36.047+0000","position":{"x": 9.42,"y": 15.75,"z": 3.00},
"confidence":50.00}
{"message_id":1467,"epc_id":"3175050952000030de000001","timestamp":"2018-07-20T10:47:40.167+0000","position":{"x": 10.12,"y": 10.82,"z": 3.00},
"confidence":50.00}
{"message_id":1468,"epc_id":"3175050952000030de000000","timestamp":"2018-07-20T10:47:40.203+0000","position":{"x": 12.24,"y": 13.65,"z": 3.00},
"confidence":50.00}
{"message_id":1469,"epc_id":"3175050952000030de000002","timestamp":"2018-07-20T10:47:41.198+0000","position":{"x": 9.42,"y": 15.75,"z": 3.00},
"confidence":50.00}
{"message_id":1470,"epc_id":"3175050952000030de000000","timestamp":"2018-07-20T10:47:45.068+0000","position":{"x": 12.24,"y": 13.65,"z": 3.00},
"confidence":50.00}
{"message_id":1471,"epc_id":"3175050952000030de000001","timestamp":"2018-07-20T10:47:45.069+0000","position":{"x": 10.12,"y": 10.82,"z": 3.00},
"confidence":50.00}
{"message_id":1472,"epc_id":"3175050952000030de000002","timestamp":"2018-07-20T10:47:46.111+0000","position":{"x": 9.42,"y": 15.75,"z": 3.00},
"confidence":50.00}
```

Stopping the Simulation

To stop simulation:

1. Change to the rtl_s directory using the following command:

```
cd rtl_s
```


2. Stop RTLS using the following command:

```
./rtls.sh -sk stop
```



The user running `rtls.sh` must be in the Docker group. If the user is not in the Docker group, use `sudo` when running `rtls.sh`.

Running CLAS with ATR7000 Hardware

Bringing the RTLS System Live with ATR7000 Readers

After validating the CLAS software installation using simulators, configure the CLAS software to run with ATR7000 readers.

For more information on running the CLAS software with ATR7000 readers, see the Post-Installation ZAATS Validation section in the ZAATS Deployment Guide.

