MotionWorks Enterprise

2.0

Installation Guide

P1123394-05EN

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.

Publication Date

October 25, 2024

Contents

Contents

Contents

Getting Started

Introduction

This guide provides instructions for installing MotionWorks Enterprise (MWE) 2.0 software from Zebra Technologies Corporation, and specifications for servers hosting the software. For additional information, refer to the *MWE 2.0 Configuration Guide* and *MWE 2.0 User Guide*.

Zebra Technologies offers world-class real-time asset tracking and management software solutions to optimize the flow of goods in complex logistical operations, increasing productivity, lowering operational costs, and improving safety and security. A wide range of scalable Real Time Locating Systems (RTLS) technologies generate accurate, on-demand information about asset location and status.

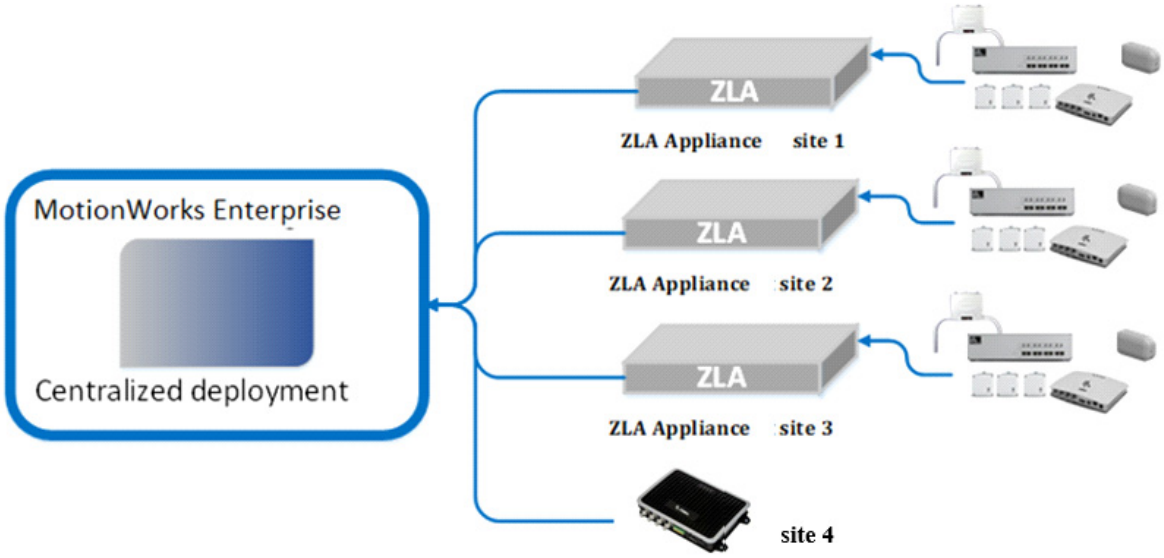
The MWE software suite provides tools for designing, configuring, operating, and troubleshooting RTLS solutions. MWE serves as the central repository for all real-time location and communication data captured by the RTLS tracking infrastructure, and integrates this data with customer and third-party applications.

Some location and telemetry RFID technologies supported by MWE include:

- Passive RFID
- UWB
- Bluetooth Low-Energy
- Wi-Fi
- ISO 24730
- GPS

The following diagram provides an overview of RTLS system components and data flow. Sensors connected to a network, wired or wirelessly, detect over-the-air RF transmissions from tags (RF transmitters), and the generated data flows across the network to a Zebra Location Appliance (ZLA) and then to the MWE server. Some sensors, such as passive RFID readers, send data directly to the MWE server without requiring a ZLA. Multiple sites are supported, each with its own ZLA.

The ZLA can be virtual or physical. A physical ZLA is a 1U rackmount box typically installed on-site. Depending on the RTLS technology deployed, a ZLA can be installed on-premises at a site, or in a remote data center. A ZLA runs location algorithms and feeds tag, blink, and location data to an MWE server. The ZLA software also includes filters for removing redundant data and reducing network traffic. The MWE server stores and retrieves data from a database, and forwards data and events to external systems.



A virtual ZLA is provided as an OVA file with pre-loaded software ready to deploy, or via a ZLA installer that converts a Linux virtual server into a ZLA appliance. Supported Linux flavors include Red Hat 7.9/8.x/9.x, Ubuntu 20.04 LTS, and Ubuntu 22.04 LTS.

MWE 2.0 also supports direct communication from passive RFID readers to MWE running locally or in the cloud. No ZLA appliance is required in this case.

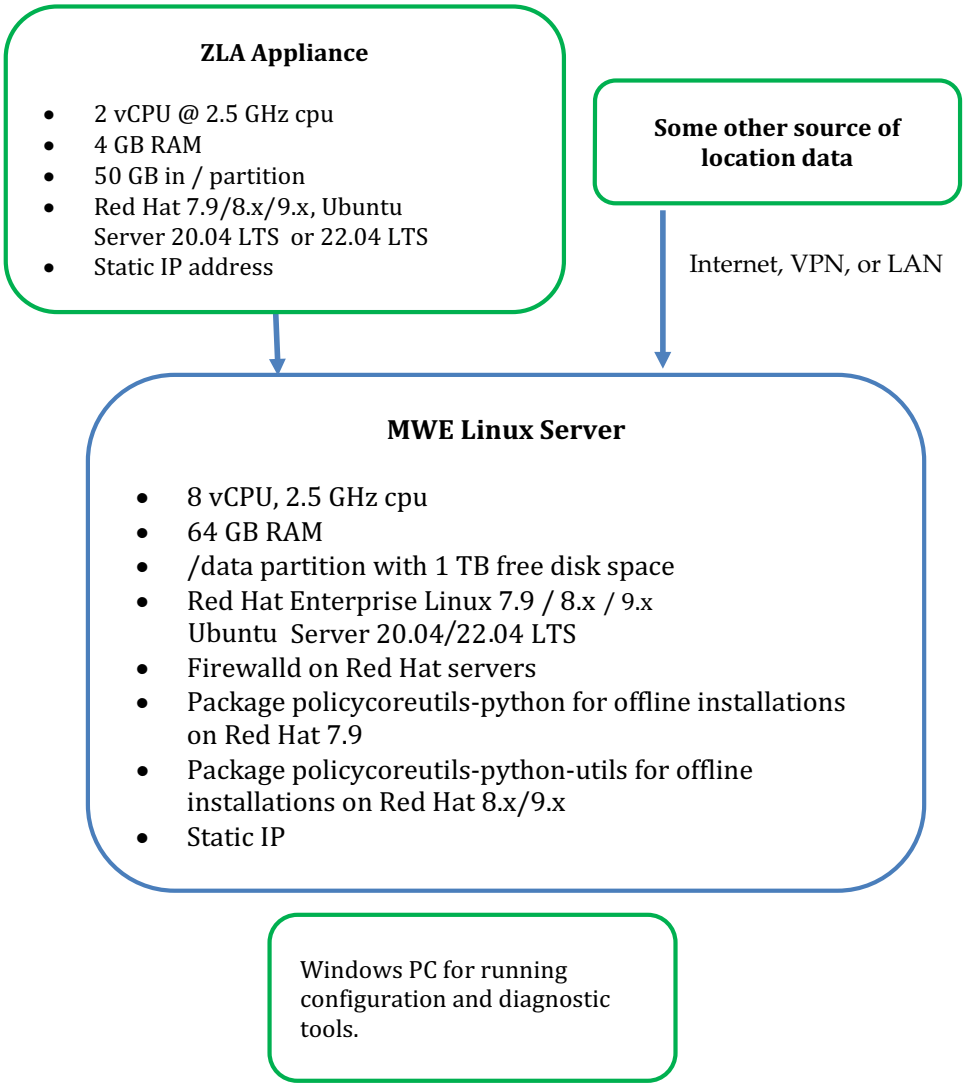
Minimum Server Specs

Location data flows to the MWE servers from a ZLA device or another source. MWE deployment requires at least one Linux server. A ZLA appliance may or may not be required. Following are the minimum hardware and software requirements for three types of deployments. The user selects one of these deployment types when running the MWE installation script.

Production Deployment

Full functionality for production deployments.

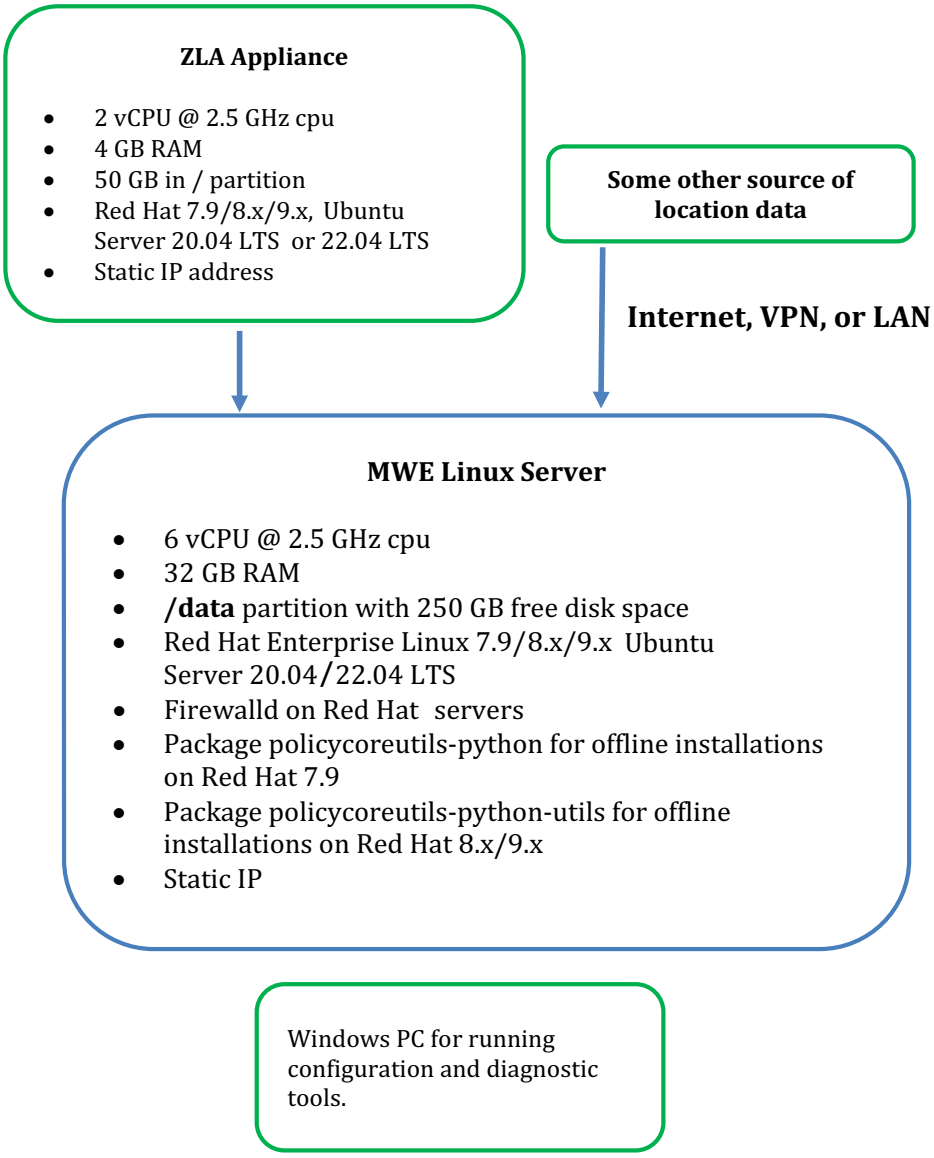
Figure 1 Production Deployment - Full Functionality



Demo Deployment

Full functionality for deployments with a light load.

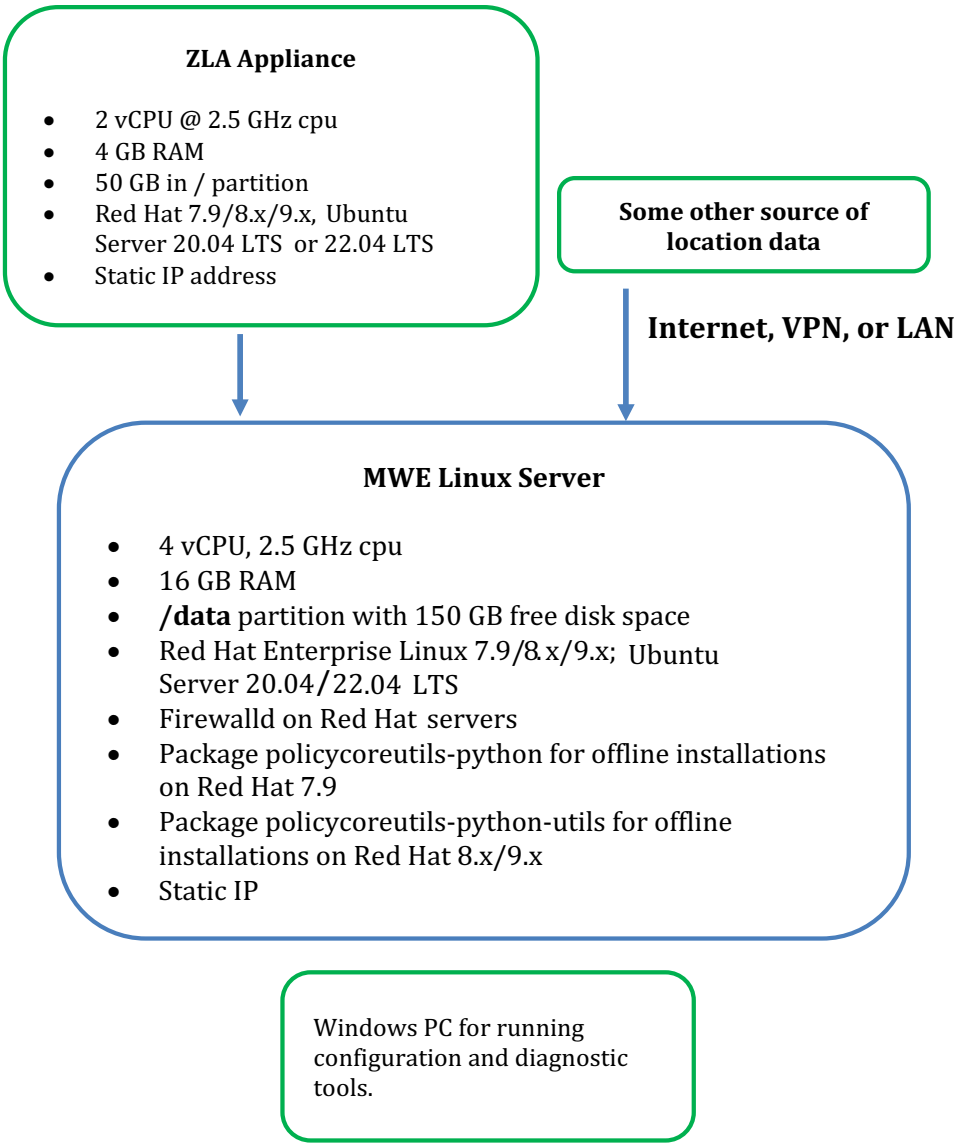
Figure 2 Demo Deployment - Full Functionality



Mimimum Demo Deployment

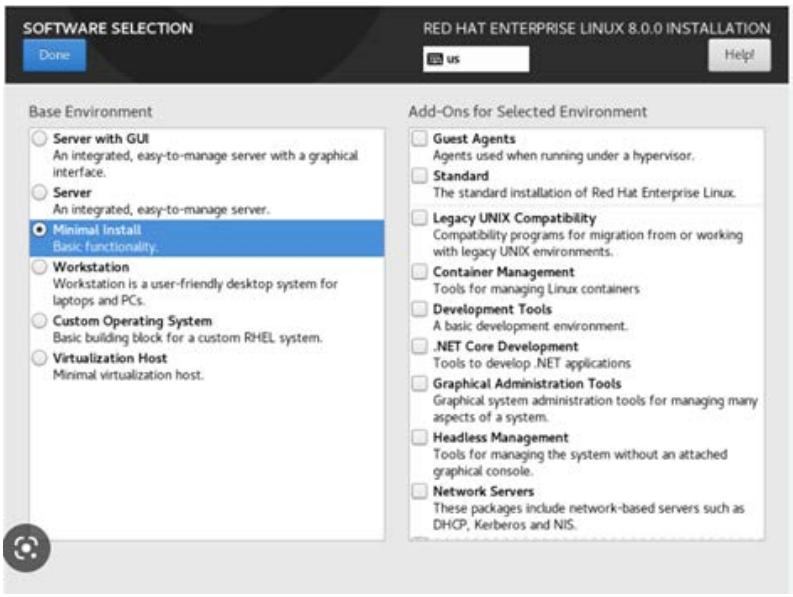
In demo deployments with minimal load the following functionality is turned off: historical service logs, Kibana tool, and MWE monitoring of system health. Historical application data, such as tag blink and event history, is still available.

Figure 3 Minimal Demo Deployment - Most Functionality



Notes on Linux Server Requirements

- Installation of MWE 2.0 on a Linux server requires a **/data** partition located directly under the **/root** directory.
- For Red Hat servers hosting MWE 2.0:
  - The **Minimal Install** option for the operating system is sufficient.



  - Install and run the firewalld daemon on the server before installing the MWE software.
  - Package **`policycoreutils-python`** must be installed on Red Hat 7.9.
    Package **`policycoreutils-python-utils`** must be installed on Red Hat 8.x/9.x.

    To verify this, run the command:
    `# rpm -q policycoreutils-python` or **`# rpm -q policycoreutils-python-utils.`**

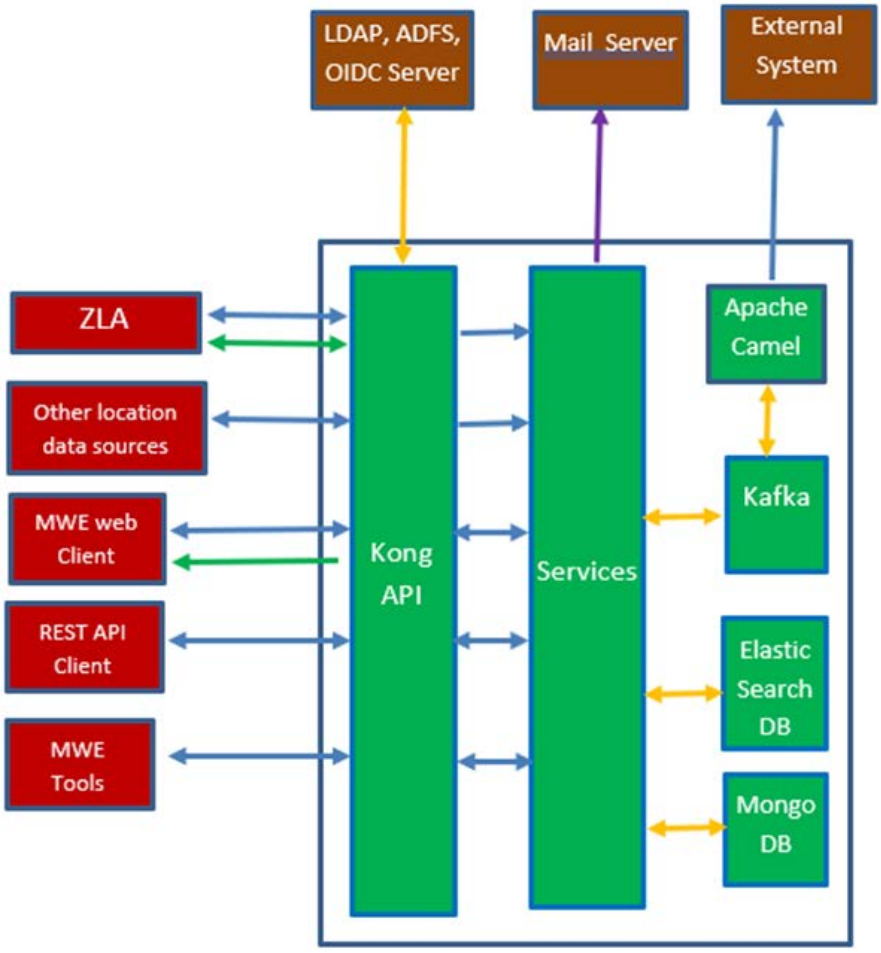
    If the server has connection to a local or public yum repository, to install this package, run the command:
    `# yum install policycoreutils-python` or **`# yum install policycoreutils-python-utils`**
- For an Ubuntu server hosting MWE or ZLA software, the Ubuntu Server edition is required.

The core MWE services on the Linux server are installed as Docker containers running on a local Docker subnet created by the installation script. This adds the following two requirements:

- ipv4 forwarding must be enabled on the server.
- If a security agent is installed on the Linux server, it must allow all traffic within the Docker subnet and between the Docker subnet and the local host.

MWE Software Components

The following reference diagram illustrates the various MWE software components hosted on the MWE Linux server and their relationship within the MWE software:



MWE Linux Server

NOTES: The arrows in the figure indicate the following:

Blue arrows	http/https connections
Orange arrows	TCP connections
Green arrow	WebSocket connection
Purple arrow	SMTP connection

- MWE Tools include System Builder and other tools.
- All core MWE services run on the Linux server as Docker containers.

Installation Files

Obtain a link to download the MWE installation files from Zebra detailed below. In the filenames, ‘n’ denotes the latest 2.0 release version available, and 'm,' when present, denotes the installer version. A new installer may occasionally be available to consolidate hotfixes and avoid having to apply them manually after installation.

mwe-containers-setup-2.0.n.m-offline.tar.gz	Installs/upgrades MWE core services on a Linux server. The installer is self-contained and does not require Internet connection.
mwe-containers-setup-2.0.n.m-online.tar.gz	Installs/upgrades MWE core services on a Linux server. This installer is much smaller than its offline counterpart but requires the Linux server to have Internet access upon installation.
mwe_tools_2.0.n.exe	Installs/upgrades a set of MWE configuration and diagnostic tools on any Windows PC. These tools can connect to the MWE (Linux) server or to different types of location sensors to perform configuration or diagnostics tasks. See Network Connections and Ports to determine what network ports must be opened.
ZLA-2.0.n-m.i386.rpm	Installs or upgrades the software on a Red Hat ZLA appliance. See Installing ZLA Software for details.
ZLA-2.0.n-m.i386.deb	Installs or upgrades the software on an Ubuntu ZLA appliance. See Installing ZLA Software for details.

Installing MWE

This section describes off-line MWE installation for Linux servers with no access to the Internet, and optional on-line installation for servers with Internet access.

Off-Line Installation

Off-line installation does not require the Linux server to have Internet access. Installation involves loading a **tar.gz** file of several GB to the Linux server, so copying it prior to a scheduled installation is recommended. Before installation, verify the MWE Linux Server requirements in Minimum Server Specs on page 7 are met.

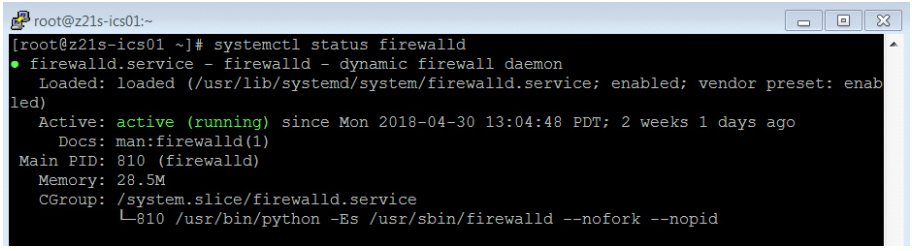
Initial Checks

Verify that the following requirements are met:

- A **/data** partition exists on the Linux server. See Minimum Server Specs on page 7.
- The package **policycoreutils-python** is installed on Red Hat 7.9. The package **policycoreutils-python-utils** is installed on Red Hat 8.x/9.x. See Notes on Linux Server Requirements on page 10 to check for and install this package.
- The firewalld daemon is installed and running on a Red Hat server. Enter the following command:

  # systemctl status firewalld

  Verify the status as **active (running)**, as in the following screen.



Copying and Extracting the Installation Package

1. Using WinSCP, Putty, or a similar SSH client, copy the **mwe-containers-setup-2.0.n.m-offline.tar.gz** installation package to the **/data** directory on the Linux server.
2. Using a Terminal window, Putty, or a similar SSH client, log into the MWE server using the **root** account or an account with adequate permissions.
3. Change directory to **/data** and extract the **tar.gz** file:

```
# cd /data
# tar -xvf  mwe-containers-setup-2.0.n.m-offline.tar.gz
```

NOTE: Occasionally characters are misinterpreted when copying a command from a document and pasting it into a Putty or Terminal window. Some commands may require manual entry.

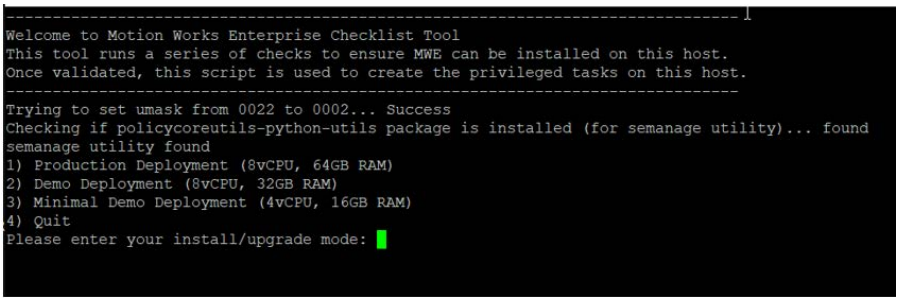
Performing Environment Validation and Initial Setup

The following setup command verifies the environment meets requirements. The setup script exits and displays a message if all requirements are not met. The setup command also installs the Docker service, creates an mwe user account, and extracts files and docker images.

1. Using the root account or an account with sufficient permissions to run the **mwe_setup_admin.sh** command, extract and run the following commands:

```
# cd /data/mwe_setup
# ./mwe_setup_admin.sh --offline-setup
```

2. Select one of four installation options.



- **Production Deployment** installs full functionality and is for production deployments. The installation script checks the following and exits if one of these conditions is not met:
  - Red Hat 7.9/8.x/9.x or Ubuntu 20.04 or Ubuntu 22.04 is installed
  - There is at least 64 GB of memory installed
  - There is at least 1000 GB (1 TB) of disk space available under **/data**
  - firewalld is running on a Red Hat server
- **Demo Deployment** includes full functionality, but is meant for demonstrations or proof of concepts with a light load. The installation script checks the following and exits if one of these conditions is not met:
  - Red Hat 7.9/8.x/9.x or Ubuntu 20.04 or Ubuntu 22.04 is installed
  - There is at least 32 GB of memory installed
  - There is at least 250 GB of disk space available under **/data**

- firewalld is running on a Red Hat server

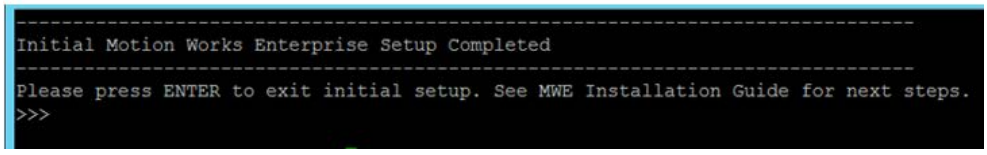
The display recommends 8 vCPU's, but the setup script does not enforce it. You can use 4 vCPU for a demo deployment with a light load if occasional response time delays are acceptable.

- **Minimal Demo Deployment** includes most functionality and is for deployments with minimal load. The following functionality is turned off: Camel, Resource Alerts, historical service logs, Kibana tool, and WherePort Health alerts. Historical application data, such as tag blink and event history, is still available. To enable/disable services after installation, contact Zebra Support.

  The installation script checks the following and exits if one of these conditions is not met:

  - Red Hat 7.9/8.x/9.x or Ubuntu 20.04 or Ubuntu 22.04 is installed
  - There is at least 16 GB of memory installed
  - There is at least 150 GB of disk space available under `/data`
  - firewalld is running on a Red Hat server
- **Quit**

It may take a couple of minutes for the setup script to extract all the files. If the setup script is successful, the following screen displays:



3. Press **Enter** to exit setup and continue MWE software installation.

Installing MWE Software

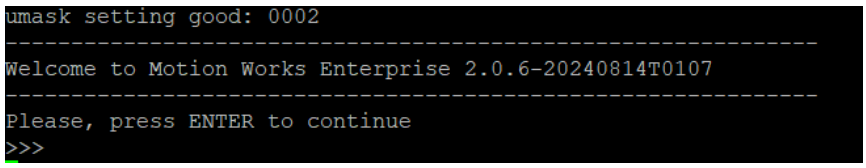
1. Switch to the mwe account (default password is Zebra123):

   ```
   # su - mwe
   ```

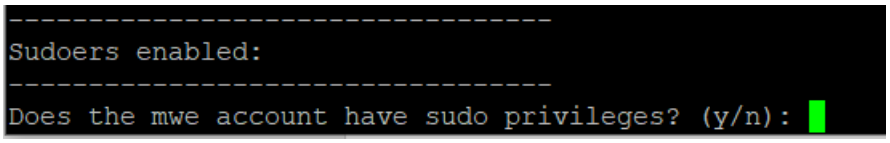
2. Change directory and run the installation script:

   ```
   $ cd /data/mwe
   $ ./mwe --offline-install
   ```

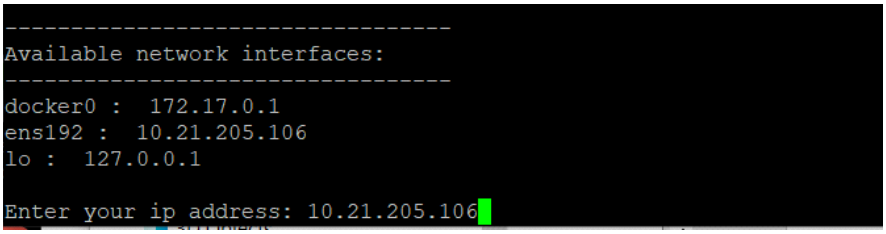
3. When prompted, press **Enter**.



4. When asked if the newly created mwe account has sudo privileges, enter **n** (typical response). Enter **y** only if mwe account permissions were manually modified.

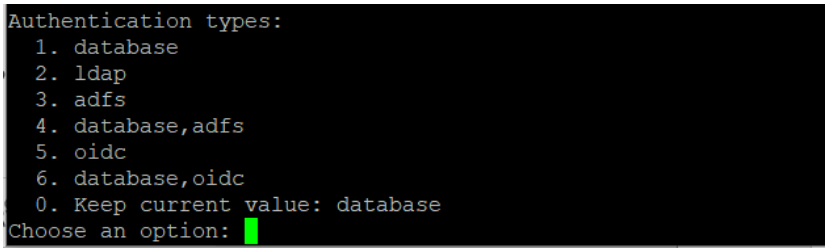


5. Enter the IP address of the Linux server where this installation is being done. This is one of the network interfaces listed by the installation script as shown.

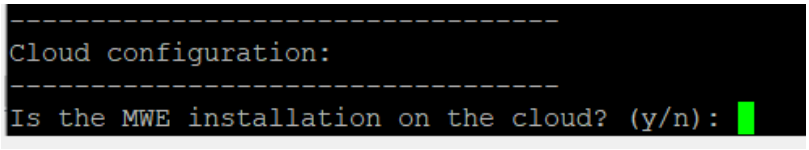


6. Enter the number of sites this MWE installation will support. If you are not sure, enter **1**. MWE automatically adjusts this value as needed.

7. Select an authentication type. At installation, select option **1. database**. After installation, you can choose and configure a different authentication type, as explained in the *MWE 2.0 Configuration Guide*.

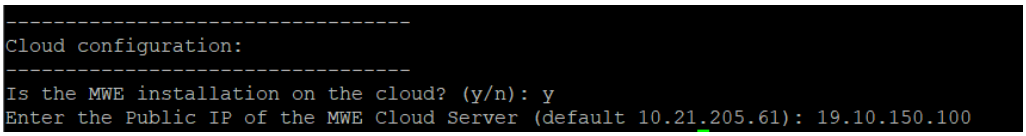


8. Respond to **`Is the MWE installation on the cloud? (y/n)`** based on the MWE deployment.

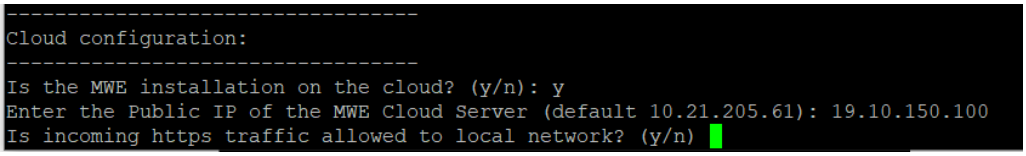


- Enter **n** if the MWE server is hosted in a private network.
  Specifically, the MWE server and any reader/location sensors sending data to MWE are all on the same network, or on different subnets linked by a router. The MWE server and the readers/sensors can reach each other's IP addresses and traffic between them can flow unimpeded (after opening necessary ports. See Network Connections and Ports).
- Enter **y** if the MWE server is hosted in a cloud and has a public IP address.
  Specifically, the MWE server is in a cloud behind the cloud firewall while the readers/sensors are on-premises at a site behind a local firewall.

  If you respond **y** (yes), enter the public IP address of the MWE server in the cloud.

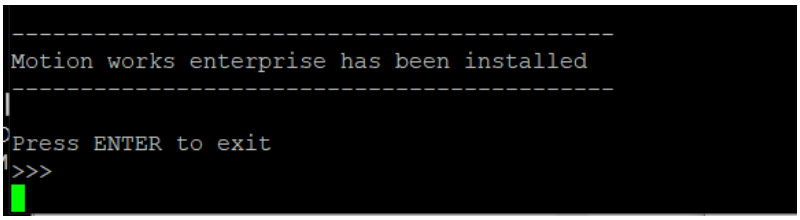


9. When prompted **`Is incoming https traffic allowed to local network? (y/n)`** the response depends on connections initiated by the MWE server in the cloud to the readers/sensors in the local network.
   - Answer no **`(n)`** if the MWE server in the cloud cannot directly reach the RFID readers/sensors IP addresses on the local network because the reader IP is private and sits behind the site's firewall. Because there is no public IP for the reader/sensor, HTTPS traffic from the MWE server in the cloud to a local reader is not allowed.
   - Answer yes **`(y)`** if the local RFID readers/sensors have a public Internet address, or otherwise the MWE server is in a private cloud that allows the MWE server to initiate a connection to the reader IP address.



   The installation script loads the Docker images, which typically takes several minutes.

10. When installation completes, press **Enter** to exit the installation.

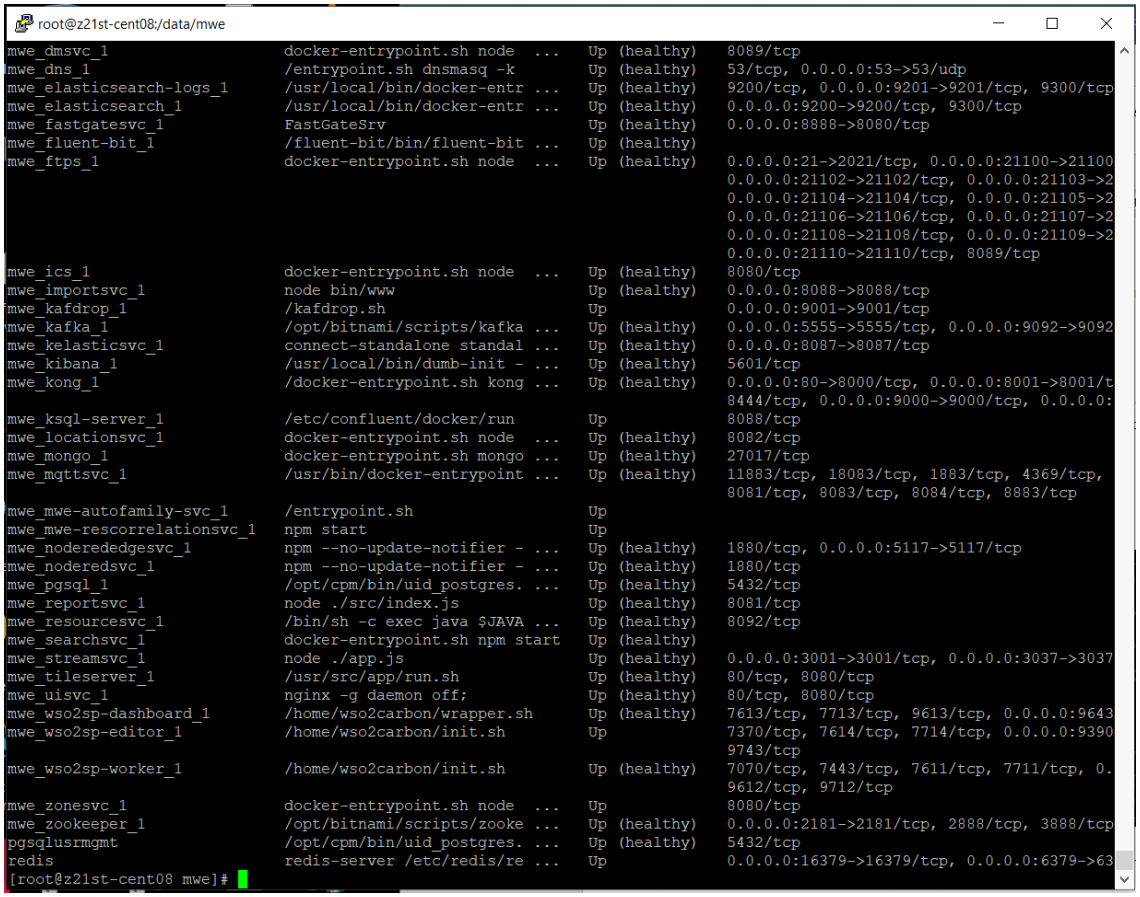


Checking Status

Verify that all MWE services are operational. Depending on server resources (CPU, memory), this may take a few minutes. MWE services run as Docker containers which can be stopped and started. Use the **docker** and **docker-compose** commands to check the status of containers.

1. To check the status of the MWE services, run the following commands:

```
$ cd /data/mwe
$ docker-compose ps
```



2. Verify the **`State`** column displays **`Up`** for all containers. Some services also report a health condition and should show **`healthy`**.

NOTE: You may see several warnings if you run the docker-compose command under the **root** account. Ignore these warnings. If you switch to the mwe account, the warnings do not display.

```
# su - mwe (if prompted, the default password is Zebra123)
$ cd /data/mwe
$ docker-compose ps
```

To switch back to the root account use:

```
$ su -
```

To restart all MWE services for any reason, either reboot the server or use these commands:

```
# su - mwe (if prompted, the default password is Zebra123)
$ cd /data/mwe
$ ./mwe --restart
```

MWE Files Location

The MWE files are copied to the following directories on the Linux server:

Installation package:	`/data/`
Setup files:	`/data/mwe_setup/`
MWE commands and configuration (.env):	`/data/mwe/`
MWE Services configuration:	`/data/mwe-conf/`
Docker images and containers:	`/data/docker/`
3rd Party services and database backups:	`/data/zebra`
Log files:	`/data/mwe-logs`

Connecting RFID Readers

This section applies only to new installations that deploy Zebra passive RFID readers. Add these readers, including FX7500, FX9600, FXR90, and ATR7000, to MWE in the Device Manager page in the MWE web client. Refer to the *MWE Device Manager User Guide*.

Note the following:

- MWE 2.0.5 and earlier releases use non-secure communication with RFID readers by default, and therefore a reader added in Device Manager can communicate with the MWE server regardless of what SSL certificate is installed on the server. By contrast, MWE 2.0.6 and later support only secure connection between readers and the MWE server, and therefore a reader validates the server SSL certificate before connecting.
- All versions of MWE are installed with a self-signed certificate using the domain name **zebramwe**. This certificate is accepted by web browsers but display the **Not secure** warning in the URL bar. However, in MWE 2.0.6 and later, a reader by default does not accept this certificate.
- Therefore, in MWE 2.0.6 or later, to allow FX/ATR readers to communicate with the MWE server, perform one of the following options (refer to the *MWE Configuration Guide* for instructions):
  - Install a valid CA SSL certificate on the MWE server
  - Install a self-signed certificate using the FQDN (Fully Qualified Domain Name) of the MWE server
  - Configure the MWE server to instruct readers to accept the default **zebramwe** domain certificate.

On-Line Installation

The following installation procedure assumes that, during installation, the MWE Linux server has access to the Internet. If this is not the case, see Off-Line Installation. Before installation, verify that the requirements listed in Minimum Server Specs on page 7 for the MWE Linux Server are met.

Performing Initial Checks

Before installing, verify the following requirements are met:

- A **/data** partition exists on the Linux server. See Minimum Server Specs on page 7.
- The MWE Linux server has access to the following website:

  https://registry.zebramwe.com

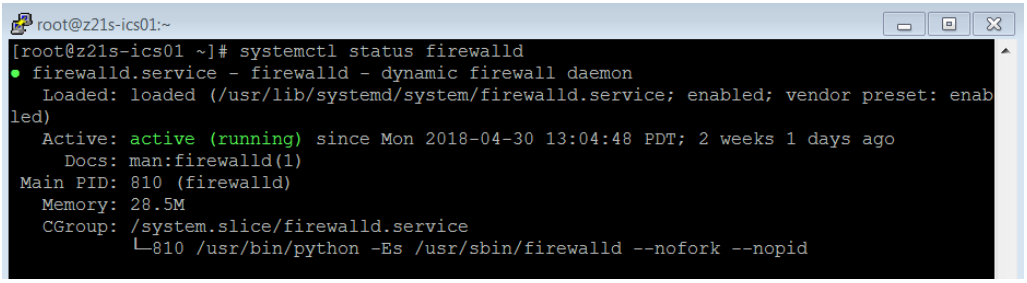
  To check access, use the curl command, for example:

  ```
  # curl -k https://registry.zebramwe.com
  ```

  This returns no error if the server has access to https://registry.zebramwe.com. Otherwise a connection timeout or similar error occurs.
- The firewalld daemon is installed and running on a Red Hat server. Use the following command to verify:

  ```
  # systemctl status firewalld
  ```

  The result should show the status as **active (running)**.



Copying and Extracting the Installation Package

1. Using WinSCP, Putty, or a similar SSH client, copy the **mwe-containers-setup-2.0.n.m-online.tar.gz** installation package to the **/data** directory on the MWE Linux server.
2. Using a Terminal window, Putty, or a similar SSH client, log into the MWE server using the root account or an account with sufficient permissions.
3. Change the directory to **/data** and extract the **tar.gz** file:

   ```
   # cd /data
   # tar -xvf mwe-containers-setup-2.0.n.m-online.tar.gz
   ```

NOTE: Occasionally characters are misinterpreted when copying a command from a document and pasting it into a Putty or Terminal window. Some commands may require manual entry.

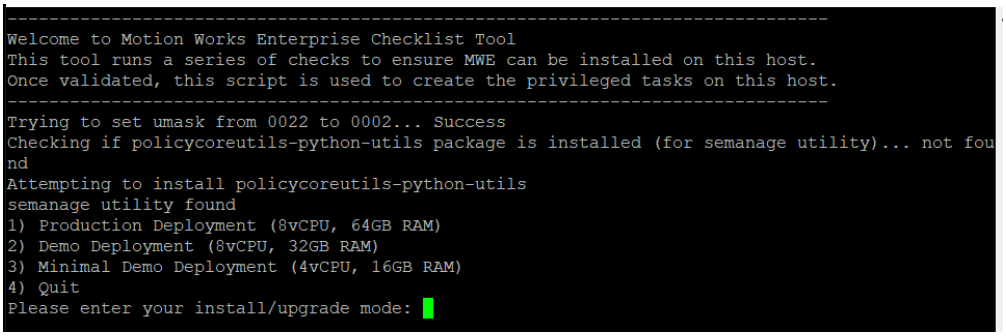
Performing Environment Validation and Initial Setup

1. Using the root account or an account with sufficient permissions to run the **`mwe_setup_admin.sh`** command, run the following to verify the environment meets requirements. if one or more requirements are not met, the setup script exits and displays a message.

   If the requirements are met, the command installs the Docker service, creates an mwe user account, and extracts files and docker images.

   ```
   # cd /data/mwe_setup
   # ./mwe_setup_admin.sh --setup
   ```

2. Select one of four installation options.



   - **Production Deployment** installs full functionality and is for production deployments. The installation script checks the following and exits if one of these conditions is not met:
     - Red Hat 7.9/8.x/9.x or Ubuntu 20.04 or Ubuntu 22.04 is installed
     - There is at least 64 GB of memory installed
     - There is at least 1000 GB (1 TB) of disk space available under **`/data`**
     - firewalld is running on a Red Hat server
     - Internet connectivity
     - Connectivity to MWE registry (https://registry.zebramwe.com)
   - **Demo Deployment** includes full functionality, but is meant for demonstrations or proof of concepts with a light load. The installation script checks the following and exits if one of these conditions is not met:
     - Red Hat 7.9/8.x/9.x or Ubuntu 20.04 or Ubuntu 22.04 is installed
     - There is at least 32 GB of memory installed
     - There is at least 250 GB of disk space available under **`/data`**
     - firewalld is running on a Red Hat server
     - Internet connectivity
     - Connectivity to MWE registry (https://registry.zebramwe.com)

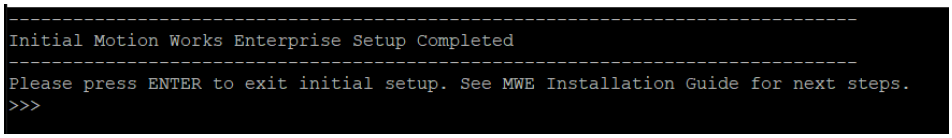
     The display recommends 8 vCPU's, but the setup script does not enforce it. You can use 4 vCPU for a demo deployment with a light load if occasional response time delays are acceptable.

- **Minimal Demo Deployment** includes most functionality and is for deployments with minimal load. The following functionality is turned off: Camel, Resource Alerts, historical service logs, Kibana tool, and WherePort Health alerts. Historical application data, such as tag blink and event history, is still available. It is possible to enable/disable services after installation; contact Product Support.

  The installation script checks the following and exits if one of these conditions is not met:

  - Red Hat 7.9/8.x/9.x or Ubuntu 20.04 or Ubuntu 22.04 is installed
  - There is at least 16 GB of memory installed
  - There is at least 150 GB of disk space available under **`/data`**
  - firewalld is running on a Red Hat server
  - Internet connectivity
  - Connectivity to MWE registry (https://registry.zebramwe.com)
- **Quit**

If all checks are successful, the following screen displays:



3. Press the **Enter** key to exit the setup and continue with MWE software installation.

Installing the MWE Software

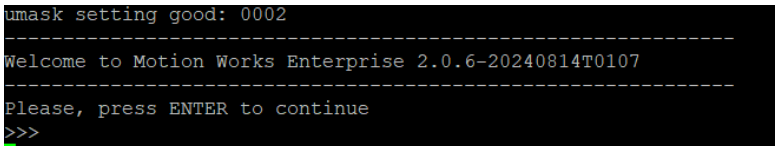
1. Switch to the mwe account (default password is Zebra123):

   ```
   # su - mwe
   ```

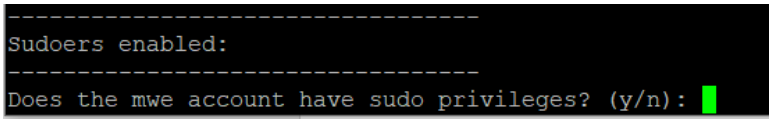
2. Change directory and run the installation script:

   ```
   $ cd /data/mwe
   $ ./mwe --install
   ```

3. Press **Enter** to continue.



4. Enter login credentials for https://registry.zebramwe.com to download the MWE Docker images. Obtain the login credentials from Zebra.
5. When asked if the newly created mwe account has sudo privileges, enter **n** (typical response). Enter **y** only if mwe account permissions were manually modified.



6. Enter the IP address of the Linux server where this installation is being done. This is one of the network interfaces listed by the installation script.

```
---------------------------------
Available network interfaces:
---------------------------------
docker0 :  172.17.0.1
ens192 :  10.21.205.106
lo :  127.0.0.1

Enter your ip address: 10.21.205.106
```

7. Enter the number of sites the MWE installation will support. If you are not sure, enter **1**. MWE automatically adjusts this value as needed.

```
---------------------------------
Num sites
---------------------------------
Enter the number of sites this MWE installation will support: (default 1):
```

8. Select an authentication type.

   At installation, select option **1. database**. After installation, you can choose and configure a different authentication type, as explained in the *MWE 2.0 Configuration Guide*.

```
Authentication types:
  1. database
  2. ldap
  3. adfs
  4. database,adfs
  5. oidc
  6. database,oidc
  0. Keep current value: database
Choose an option:
```

9. Respond to **Is the MWE installation on the cloud? (y/n)** based on the MWE deployment:

   - Enter **n** if the MWE server is hosted in a private network.
     Specifically, the MWE server and any reader/location sensors sending data to MWE are all on the same network, or on different subnets linked by a router. The MWE server and the readers/sensors can reach each other's IP addresses and traffic between them can flow unimpeded (after opening necessary ports. See Network Connections and Ports).
   - Enter **y** if the MWE server is hosted in a cloud and has a public IP address.
     Specifically, the MWE server is in a cloud behind the cloud firewall while the readers/sensors are on-premises at a site behind a local firewall.

     If you answer yes **(y)**, enter the public IP address of the MWE server in the cloud.

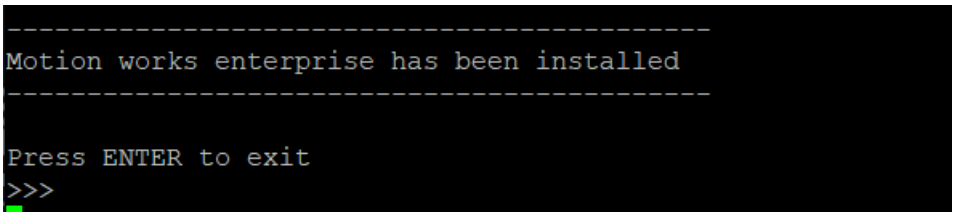
```
---------------------------------
Cloud configuration:
---------------------------------
Is the MWE installation on the cloud? (y/n): y
Enter the Public IP of the MWE Cloud Server (default 10.21.205.61): 19.10.150.100
```

10. When prompted **Is incoming https traffic allowed to local network? (y/n)** the response depends on connections initiated by the MWE server in the cloud to the readers/sensors in the local network.

    - Answer no **(n)** if the MWE server in the cloud cannot directly reach the readers/sensors IP addresses in the local network because the reader IP is private and sits behind the site's firewall. Because there is no public IP for the readers/sensors, HTTPS traffic from the MWE server in the cloud to a local reader is not allowed.

- Answer yes **(y)** if the local RFID readers/sensors has a public Internet address, or otherwise the MWE server is in a private cloud that allows the MWE server to initiate a connection to the reader IP address.

The installation script installs the Docker images, which can take several minutes. When installation completes, press **Enter** to exit installation.

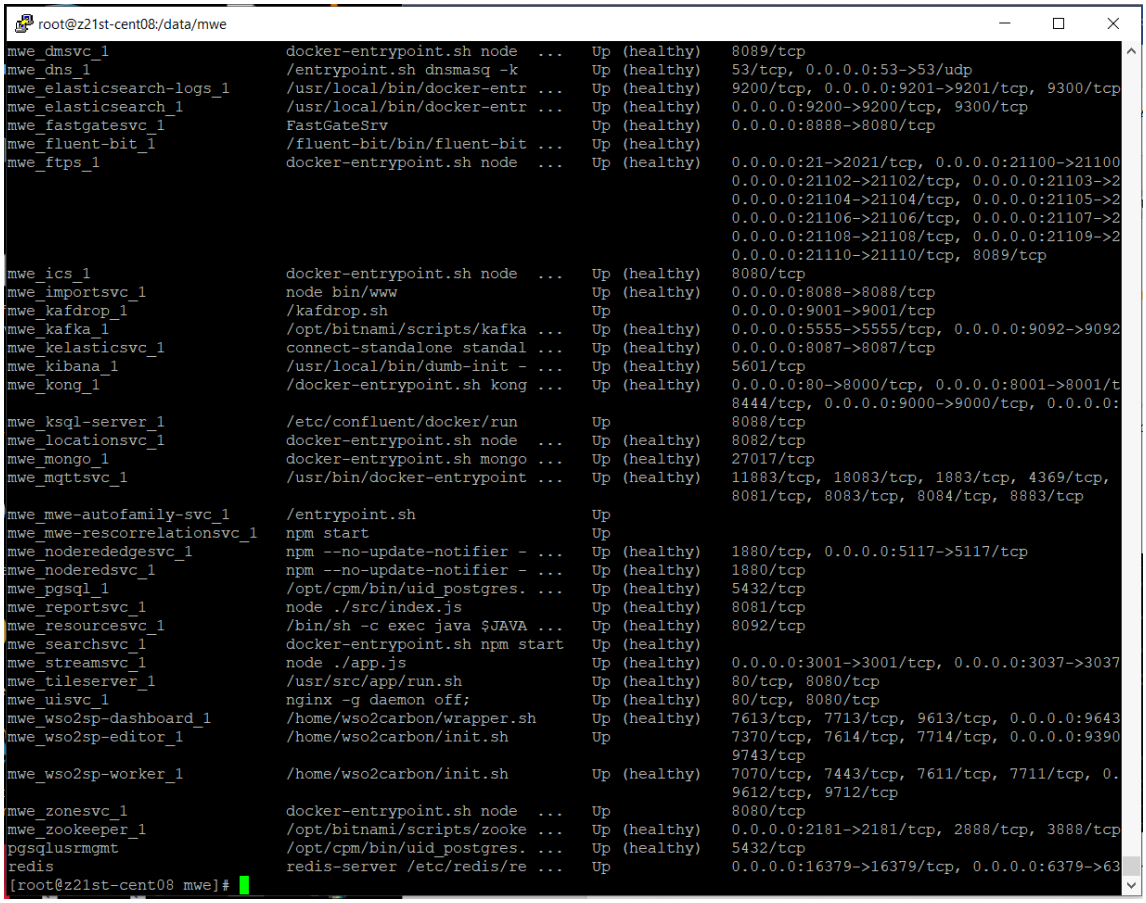


Checking Status

Verify all MWE services are operational. Depending on server resources (cpu, memory), this may take a few minutes. MWE services run as Docker containers which can be stopped and started. Use the **docker** and **docker-compose** commands to check the status of the containers.

1. To check the status of MWE services (Docker containers), run the following commands:

```
$ cd /data/mwe
$ docker-compose ps
```



2. Verify the **State** column displays **Up** for all containers. Some services also report a health condition and should show **healthy**.

NOTE: You may see several warnings if you run the docker-compose command under the root account. Ignore these warnings. If you switch to the mwe account, the warnings do not display.

```
# su - mwe (if prompted, the default password is Zebra123)
$ cd /data/mwe
$ docker-compose ps
```

To switch back to the root account use:

```
# su -
```

To restart all MWE services for any reason, either reboot the server or use these commands:

```
# su - mwe (if prompted, the default password is Zebra123)
$ cd /data/mwe
$ ./mwe --restart
```

MWE Files Location

The MWE files are copied to the following directories on the Linux server:

Installation package:	`/data/`
Setup files:	`/data/mwe_setup/`
MWE commands and configuration (.env):	`/data/mwe/`
MWE Services configuration:	`/data/mwe-conf/`
Docker images and containers:	`/data/docker/`
Third Party services and database backups:	`/data/zebra`
Log files:	`/data/mwe-logs`

Connecting RFID Readers

This section applies only to new installations that deploy Zebra passive RFID readers. Add these readers, including FX7500, FX9600, FXR90, and ATR7000, to MWE in the Device Manager page in the MWE web client. Refer to the *MWE Device Manager User Guide*.

Note the following:

- MWE 2.0.5 and earlier releases use non-secure communication with RFID readers by default, and therefore a reader added in Device Manager can communicate with the MWE server regardless of what SSL certificate is installed on the server. By contrast, MWE 2.0.6 and later support only secure connection between readers and the MWE server, and therefore a reader validates the server SSL certificate before connecting.
- All versions of MWE are installed with a self-signed certificate using the domain name **zebramwe**. This certificate is accepted by web browsers but display the **Not secure** warning in the URL bar. However, in MWE 2.0.6 and later, a reader by default does not accept this certificate.
- Therefore, in MWE 2.0.6 or later, to allow FX/ATR readers to communicate with the MWE server, perform one of the following options (refer to the *MWE Configuration Guide* for instructions):
  - Install a valid CA SSL certificate on the MWE server
  - Install a self-signed certificate using the FQDN (Fully Qualified Domain Name) of the MWE server
  - Configure the MWE server to instruct readers to accept the default **zebramwe** domain certificate.

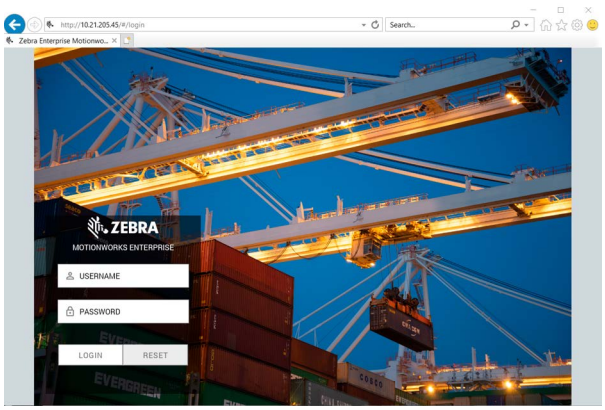
Installing MWE Tools

The **mwe_tools_2.0.n.exe** installation program included with the MWE distribution files installs a set of MWE configuration and diagnostic tools on any Windows PC. These tools can connect to the MWE server, to the ZLA, and to different types of location sensors to perform configuration, diagnostics, and troubleshooting tasks. See Network Connections and Ports for ports that must be open.

To install the MWE tools, copy **mwe_tools_2.0.n.exe** to a Windows machine, right-click on the file and select **Run as Administrator**. Follow the on-screen instructions.

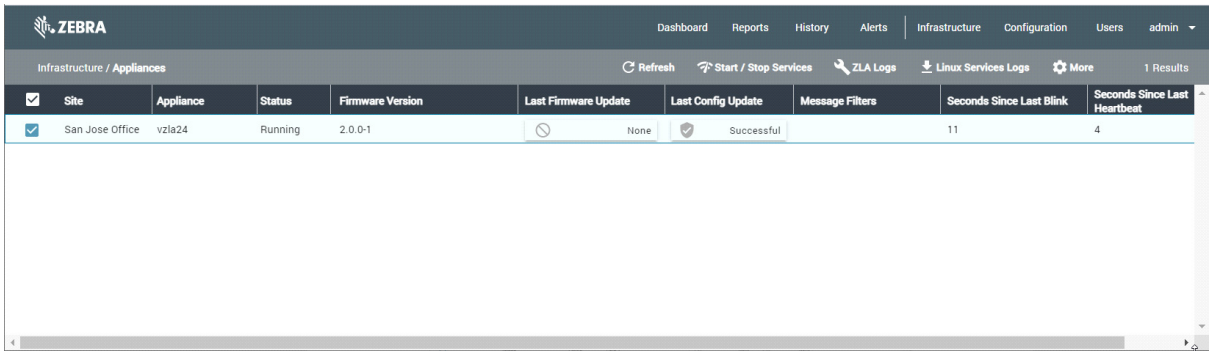
Launching the Web Client

1. Open a web browser (Chrome, Edge, or Firefox) on a client machine or server on the network, and enter http://*MWE_Server_Name* or https://*MWE_Server_Name* where *MWE_Server_Name* is the MWE Linux server name or IP address.
2. On the login page enter the **Username** and **Password** (the default is **admin** / **admin**). In MWE 2.0.6 and later, requests to http are automatically re-directed to https, and you are prompted to change the default password the first time you login.



NOTE: See the *MWE Configuration Guide* for information on adding an SSL certificate on the MWE server.

3. Open the **Infrastructure** > **Appliances** page to view the previously registered ZLA.



The **Status** column displays **Failed** (or **Activating**) until a `site.json` configuration file is published using the System Builder tool (see the *MWE Configuration Guide* for details). If your ZLA includes a default `site.json` file, the **Status** column may show **Running**, or alternate between **Running** and **Offline** if the ZLA software has not been upgraded to version 2.0.0.

Installing a World Map

After launching the web client, open the **Infrastructure > Site Manager** page for a default background world map. To optionally load a more detailed background map, for example a map with street-level information for the U.S. or other countries:

1. Locate the map in mbtiles format (extension **`.mbtiles`**). https://openmaptiles.com/downloads/planet/ offers maps with street-level information for different regions of the world.

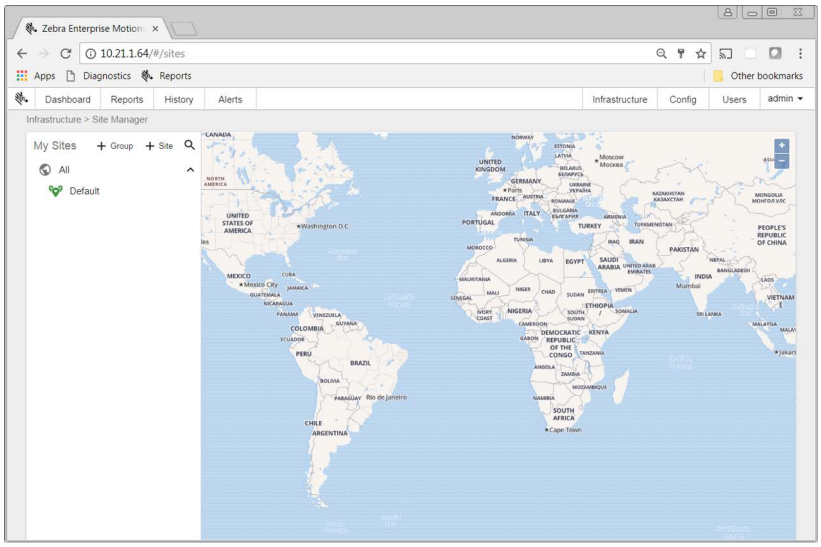
   For example, the file **`north-america_us.mbtiles`** contains U.S. street-level information. Note that downloading this large file (7 GB) may take some time.

2. Assuming you have installed MWE under **`/data`**, copy the map file to **`/data/zebra/mwe/3rdParty/tileserver-data`** on the Linux server, then delete or move the previous world map from this folder. The tileserver service, which processes the file, reads only one file from this folder.

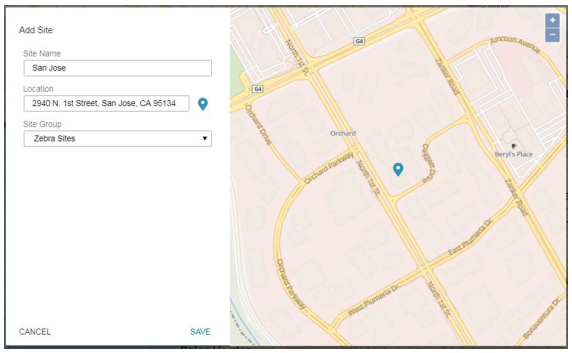
3. To restart the tileserver service on the MWE server, run the commands:

   ```
   # cd /data/mwe          (assuming you have installed MWE under /data)
   # docker-compose stop tileserver
   # docker-compose start tileserver
   ```

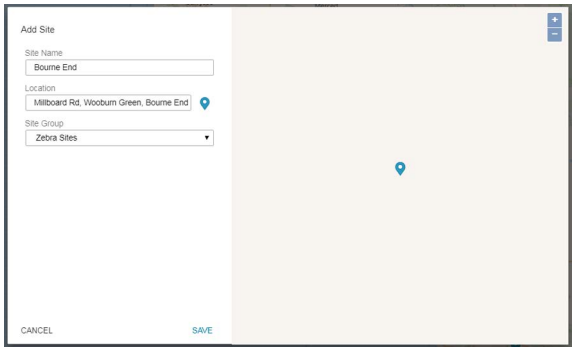
4. Relaunch the web client. Initially, the **Infrastructure > Site Manager** page may take 30- 60 seconds to display the more detailed **north-america_us.mbtiles** world map.



When adding a site, the **Add Site** window displays street level details of the US address you entered.



For other regions of the world, no map displays in the **Add Site** window.



If you keep the default installation world map, no map displays in the **Add Site** window for US addresses. This has no impact on MWE functionality.

Configuring MWE

After installation, several configuration tasks are required for a fully operational system. Refer to the following documents for instructions:

- *MWE Configuration Guide* - describes configuration tasks that are done only once or infrequently after installing the software, such as creating sites and site groups, uploading and calibrating site maps, defining zones and zone groups, and specifying location devices and algorithms for system use. The system is fully functional after this configuration.
- *MWE User Guide* - describes the basic functionality of the web client for end users, and includes configuration tasks that further customize the application or that are performed on a frequent basis such as adding users and user groups, defining access permissions, defining resource types, associating tags with resources, defining data filters, and configuring the various reports (columns displayed and column order).

Upgrading MWE

This section includes instructions for upgrading from MWE 1.4.x to MWE 2.0.5 or earlier 2.0.x releases. To upgrade to MWE 2.0.6 or higher, you must first upgrade to 2.0.5.

Upgrading from MWE 1.4.x to MWE 2.0

The upgrade script transfers data from the SQL database in MWE 1.4 to a mongoDB database on the MWE Linux server, but not all data. The following data is transferred and available after upgrade:

- Users, User groups, and Group permission configuration
- Contacts
- Zones and Zone groups
- Tags associated with resources
- Resource Types
- Resources
- Named resource custom fields and values. This applies to custom fields named in the Configuration > Token Replacement Settings report in the MWE web client. For example **`~Object Custom1~ = Color`** is transferred. However, a custom field that retains its default name, such as **`~Object Custom2~ = Resource Custom2`**, is not transferred since it is assumed not in use.

The following data is not transferred and not available after upgrade:

- The Zone and Zone Group columns in the Tags and Resources report are empty and the Site Name displays **Default** until the tags blink again.
- Unassigned tags. This applies to tags not associated with a resource ID.
- Business rules (resource alerts). A script is provided that saves these rules so they can be manually re-created after upgrade.
- Resource custom properties with default token names. Again, a custom field that retains its default name, such as for example **`~Object Custom2~ = Resource Custom2`**, is not transferred since it is assumed not in use.
- Assignment of recipients to system alerts. If you configured which system alerts are emailed to certain recipients, you must manually re-assign recipients to system alerts after upgrade.
- Saved/custom reports must be recreated manually. In MWE 2.0.4 and later, you can create a saved report and propagate it to other users.
- Token replacement of report and column names. This is not yet supported in MWE 2.0.
- Tag IDs that contain non-printable characters (such as Line Feed), which were introduced into the system by data import operations. This is rare.

The following data stored in the Elastic Search database on the Linux server in MWE 1.4.3 remains accessible after upgrade:

- Tag blink history
- Zone change history
- Event history

Requirements

MWE 2.0 requires more memory and advanced CPU on the Linux host than MWE 1.4. Not having sufficient memory and CPU may cause the upgrade to fail or MWE services to crash after upgrade. Review server requirements in Minimum Server Specs.

For offline upgrades, the MWE 2.0 installation script requires installing the package **`policycoreutils-python`** on a Red Hat server before upgrading, otherwise the installation script exits. An Ubuntu server does not require this package.

To determine if this package is installed run the following commands:

- On Red Hat 7.9:

  ```
  # rpm -q policycoreutils-python
  ```

- On Red Hat 8.x/9x:

  ```
  # rpm -q policycoreutils-python-utils
  ```

If the server has Internet connection, run the following commands to install this package:

- On Red Hat 7.9:

  ```
  # yum install policycoreutils-python
  ```

- On Red Hat 8.x/9x

  ```
  # yum install policycoreutils-python-utils
  ```

Pre-Upgrade Instructions

Off-line upgrade does not require the Linux server to have Internet access, but requires copying a **`tar.gz`** file to the MWE Linux server. Due to its size (close to 7 GB), consider copying the file prior to a scheduled upgrade.

Before upgrading, verify the requirements in Minimum Server Specs for the MWE Linux Server are met.

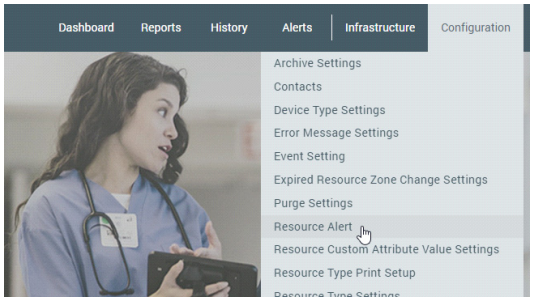
Taking a Snapshot

Take a snapshot of the MWE 1.4 Linux virtual server before upgrading so you can quickly restore the original state of the system if upgrade fails. Upgrade does not affect the MWE 1.4 Windows application server or the SQL Server host, so a snapshot is not required for these servers.

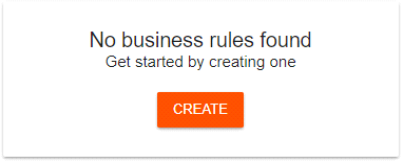
Backing Up Business Rules

The upgrade script does not preserve business rules (resource alerts) defined in MWE 1.4. The **copyBusinessRules.sh** script is provided to save the rules configuration to a file, to use after upgrade to manually add back these rules.

Verify in the Resource Alerts report if business rules (resource alerts) were defined in MWE 1.4.



If no Business Alerts exist, the following screen displays. Proceed to Stopping Services.



To save business rules:

1. Copy the **copyBusinessRules.sh** script to the **/data** directory on the Linux server.
2. Using Putty or a Terminal window, log into the MWE Linux server as root and change the file permissions to allow execution of the script:

   ```
   # cd /data
   # chmod +x copyBusinessRules.sh
   ```

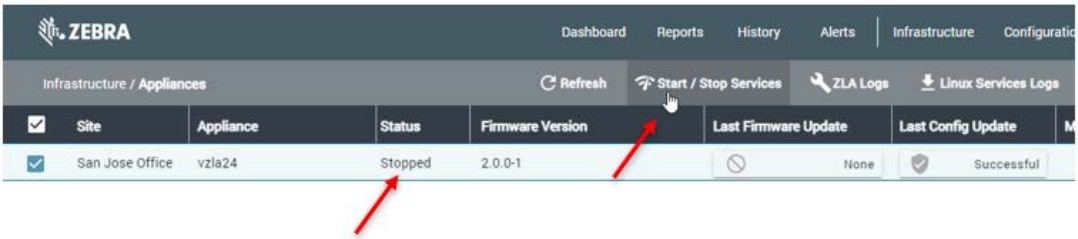
3. Run the script:

   ```
   # ./copyBusinessRules.sh
   ```

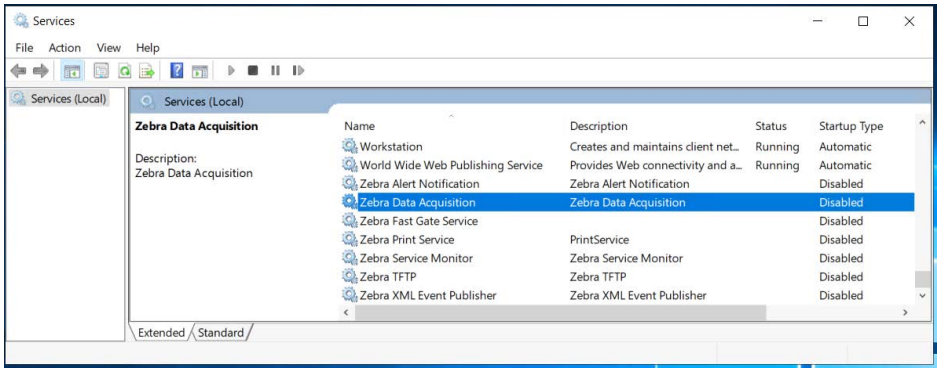
This script generates a directory **/data/saved-rules** where each existing business rule is saved as a json file.

Stopping Services

1. Stop services on ZLAs listed in the Appliances report.



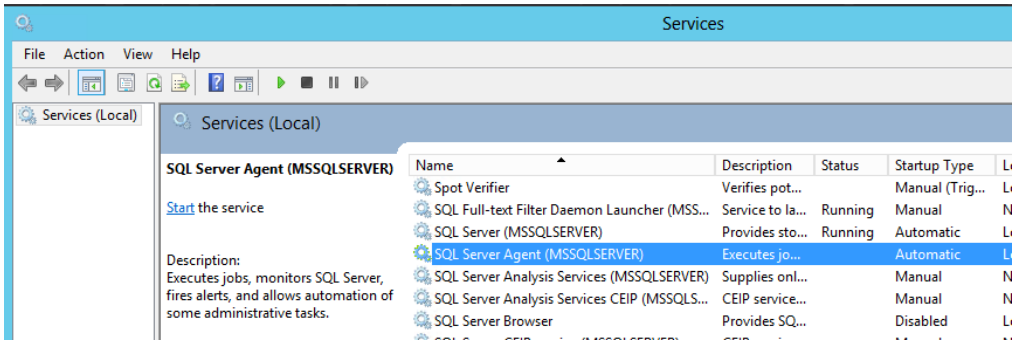
2. Using Windows Services (Service Control Manager) stop and disable all Zebra services on the MWE 1.4 Windows application service. The services that display in this screen depend on the optional modules installed on top of MWE 1.4.



3. Stop **SQL Server Agent** on the Windows server hosting SQL Server.

   IMPORTANT: Do not stop the **SQL Server** service.

   If you cannot stop SQL Server Agent because doing so would impact other non-MWE databases hosted on the same SQL Server host, disable all MWE database scheduled jobs in SQL Server Management Studio.



Upgrading

You can perform an off-line upgrade or an on-line upgrade.

- Off-line upgrade does not require the MWE Linux server to have Internet access. Because the installation package is close to 8.5 GB, copying it prior to a scheduled upgrade is recommended.
- On-line upgrade requires that the MWE Linux server have open access to the Internet during installation.

Copying and Extracting the Installation Package

1. Using WinSCP, Putty, or a similar SSH client, copy the installation package to the **/data** directory on the Linux server:
   - For an off-line upgrade, copy the **mwe-containers-setup-2.0.n.m-offline.tar.gz** installation package.
   - For an on-line upgrade, copy the **mwe-containers-setup-2.0.n.m-online.tar.gz** installation package.
2. Using a Terminal window, Putty, or another SSH client, log into the MWE server using the root account.
3. Change the directory to **/data**.

   ```
   # cd /data
   ```
4. Extract the files.
   - For off-line upgrade, enter:

     ```
     # tar -xvf  mwe-containers-setup-2.0.n.m-offline.tar.gz
     ```
   - For on-line upgrade, enter:

     ```
     # tar -xvf  mwe-containers-setup-2.0.n.m-online.tar.gz
     ```

NOTE: Occasionally characters are misinterpreted when copying a command from a document and pasting it into a Putty or Terminal window. Some commands may require manual entry.

Running the Upgrade Script

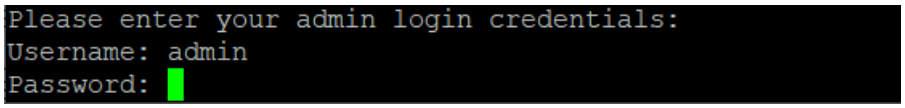
1. Change the directory.

   ```
   # cd /data/mwe_setup
   ```
2. Run the upgrade script.
   - For off-line upgrade, enter:

     ```
     # ./mwe_setup.sh --offline-upgrade
     ```
   - For on-line upgrade, enter:

     ```
     # ./mwe_setup.sh --upgrade
     ```

3. The upgrade script begins upgrading the Docker images. When prompted, enter your login credentials.



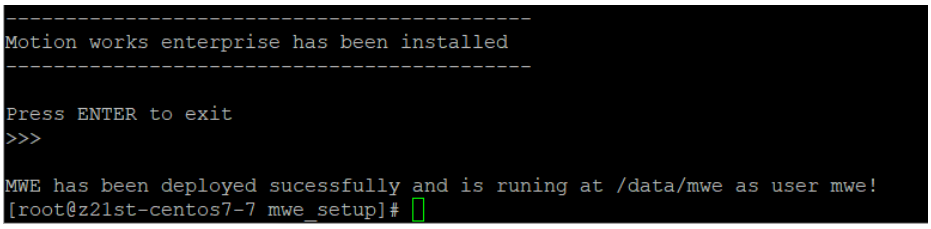
These are the same admin account login credentials used to log in to the MWE web client. Use the admin account or another account in the MWE Administrator user group.

The upgrade script uses these credentials to log into the SQL server and migrate the data in the SQL db_MWE database to a mongoDB database on the MWE server.

NOTE: No characters display when you type the password. If you enter the wrong password, you are prompted to re-enter the login credentials.

The upgrade/migration script runs for a few more minutes.

4. When the upgrade completes, press **Enter** to exit.



Running the Upgrade Again

If the upgrade script encounters an error, it posts an error message and exits without completing the upgrade. If this occurs, consult the following logs:

- Installation log: **/data/mwe_setup/mwe_setup-*timestamp*.log**
- Data migration log: **/var/log/zebra/mwe/db_migration.log**

If possible, correct the issue reported in the logs, restore the MWE 1.4 Linux snapshot, and repeat the upgrade process starting with Requirements.

If you encounter an issue you cannot resolve, see Reverting to MWE 1.4 to restore the original MWE 1.4, and contact Zebra Product Support for assistance.

Validating the Upgrade

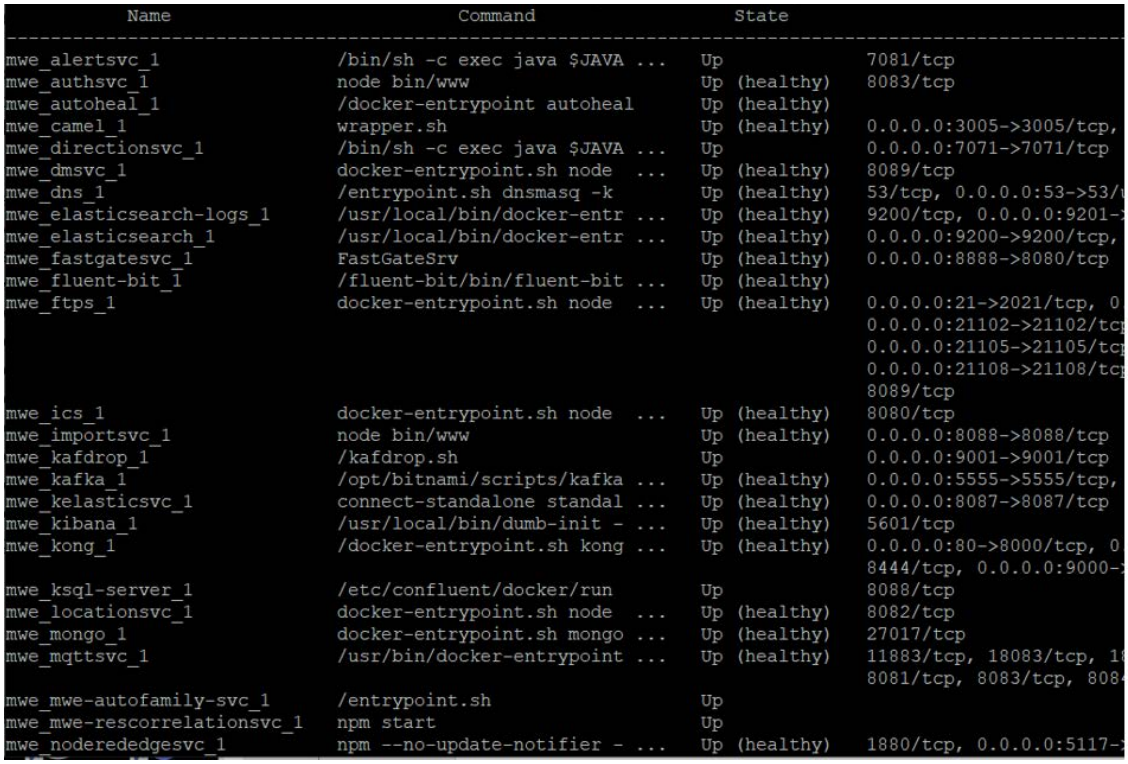
Once the upgrade script completes, verify the following:

1. Verify all MWE Services are operational by logging into the MWE Linux server using Putty or a similar SSH client and running the **`docker-compose ps`** command as **`mwe`** user.

   Depending on the resources (CPU and memory) on your server, it may take a few minutes for services to be up and running.

   ```
   # su - mwe
   # cd /data/mwe
   # docker-compose ps
   ```

Ensure the **State** column displays **Up** for all services.



```
          Name                        Command               State
-------------------------------------------------------------------------------------------------
mwe_alertsvc_1               /bin/sh -c exec java $JAVA ...   Up             7081/tcp
mwe_authsvc_1                node bin/www                     Up (healthy)   8083/tcp
mwe_autoheal_1               /docker-entrypoint autoheal      Up (healthy)
mwe_camel_1                  wrapper.sh                       Up (healthy)   0.0.0.0:3005->3005/tcp,
mwe_directionsvc_1           /bin/sh -c exec java $JAVA ...   Up             0.0.0.0:7071->7071/tcp
mwe_dmsvc_1                  docker-entrypoint.sh node  ...   Up (healthy)   8089/tcp
mwe_dns_1                    /entrypoint.sh dnsmasq -k        Up (healthy)   53/tcp, 0.0.0.0:53->53/
mwe_elasticsearch-logs_1     /usr/local/bin/docker-entr ...   Up (healthy)   9200/tcp, 0.0.0.0:9201-
mwe_elasticsearch_1          /usr/local/bin/docker-entr ...   Up (healthy)   0.0.0.0:9200->9200/tcp,
mwe_fastgatesvc_1            FastGateSrv                      Up (healthy)   0.0.0.0:8888->8080/tcp
mwe_fluent-bit_1             /fluent-bit/bin/fluent-bit ...   Up (healthy)
mwe_ftps_1                   docker-entrypoint.sh node  ...   Up (healthy)   0.0.0.0:21->2021/tcp, 0
                                                                               0.0.0.0:21102->21102/tcp
                                                                               0.0.0.0:21105->21105/tcp
                                                                               0.0.0.0:21108->21108/tcp
                                                                               8089/tcp
mwe_ics_1                    docker-entrypoint.sh node  ...   Up (healthy)   8080/tcp
mwe_importsvc_1              node bin/www                     Up (healthy)   0.0.0.0:8088->8088/tcp
mwe_kafdrop_1                /kafdrop.sh                      Up             0.0.0.0:9001->9001/tcp
mwe_kafka_1                  /opt/bitnami/scripts/kafka ...   Up (healthy)   0.0.0.0:5555->5555/tcp,
mwe_kelasticsvc_1            connect-standalone standal ...   Up (healthy)   0.0.0.0:8087->8087/tcp
mwe_kibana_1                 /usr/local/bin/dumb-init - ...   Up (healthy)   5601/tcp
mwe_kong_1                   /docker-entrypoint.sh kong ...   Up (healthy)   0.0.0.0:80->8000/tcp, 0
                                                                               8444/tcp, 0.0.0.0:9000-
mwe_ksql-server_1            /etc/confluent/docker/run        Up             8088/tcp
mwe_locationsvc_1            docker-entrypoint.sh node  ...   Up (healthy)   8082/tcp
mwe_mongo_1                  docker-entrypoint.sh mongo ...   Up (healthy)   27017/tcp
mwe_mqttsvc_1                /usr/bin/docker-entrypoint ...   Up (healthy)   11883/tcp, 18083/tcp, 1
                                                                               8081/tcp, 8083/tcp, 808
mwe_mwe-autofamily-svc_1     /entrypoint.sh                   Up
mwe_mwe-rescorrelationsvc_1  npm start                        Up
mwe_noderededgesvc_1         npm --no-update-notifier - ...   Up (healthy)   1880/tcp, 0.0.0.0:5117-
```

2. Log in to the MWE web client as **admin** and verify the data in the following reports.
   - Configuration/Contacts
   - Configuration/Zone Settings
   - Configuration/Resource Types
   - Reports/Tags
   - Reports/Resources
   - Infrastructure/Site Manager/Configure zones

NOTE: If an error message occurs when opening a report, clear your browser cache.

To restart all MWE services for any reason, reboot the server or use these commands:

```
# su - mwe (if prompted for password, the default password is Zebra123)
# cd /data/mwe
# ./mwe --restart
```

Reverting to MWE 1.4

To revert to MWE 1.4 because the upgrade script exited due to an error or another reason:

1. Restore the MWE 1.4 snapshot (Linux server).
2. Enable and start services on the MWE 1.4 Windows application server.
3. Start the SQL Server Agent service on the SQL server host.
4. In the Appliances report in the MWE web bclient, start the ZLA services.

Upgrading ZLA Software

After a successful upgrade of the MWE Linux for deployments including a ZLA, upgrade the ZLA software to a version compatible with the MWE version. See Upgrading the ZLA.

Upgrading from MWE 2.0.n to MWE 2.0.m

This section provides instructions for upgrading from MWE 2.0.n where n<5 to MWE 2.0.m where m<=5. You can perform an off-line upgrade or an on-line upgrade.

NOTE: To upgrade from MWE 2.0.5 to MWE 2.0.6, see Upgrading from MWE 2.0.5 to MWE 2.0.6.

- Off-line upgrade does not require the MWE Linux server to have Internet access during installation. Because the installation package is close to 8.5 GB, copying it prior to a scheduled upgrade is recommended.
- On-line upgrade requires that the MWE Linux server have open access to the Internet during installation.

CAUTION: if your MWE deployment uses the Camel interface to post data to an external system, note that MWE 2.0.5 includes a new version of the Camel interface (version 3.21.0) that requires a slightly different syntax in the Camel route files. **DO NOT** upgrade to 2.0.5 until Zebra Product Support reviews these files as this may break communication between Camel and external systems.

Taking a Snapshot

Following best practices, capture a snapshot of your virtual machine before upgrading, in case you need to restore the previous version.

Copying and Extracting the Installation Package

1. Using WinSCP, Putty, or a similar SSH client, copy the installation package to the **`/data`** directory on the Linux server:
   - For an off-line upgrade, copy the **`mwe-containers-setup-2.0.m-offline.tar.gz`** installation package.
   - For an on-line upgrade, copy the **`mwe-containers-setup-2.0.m-online.tar.gz`** installation package.
2. Using a Terminal window, Putty, or another SSH client, log into the MWE server using the root account.
3. Change the directory.

   ```
   # cd /data
   ```
4. Extract the files.
   - For an off-line upgrade, enter:

     ```
     # tar -xvf  mwe-containers-setup-2.0.m-offline.tar.gz
     ```
   - For an on-line upgrade, enter:

     ```
     # tar -xvf  mwe-containers-setup-2.0.m-online.tar.gz
     ```

NOTE: Occasionally characters are misinterpreted when copying a command from a document and pasting it into a Putty or Terminal window. Some commands may require manual entry.

Running the Upgrade Script

1. Change the directory.

   ```
   # cd /data/mwe_setup
   ```

2. If upgrading from 2.0.1, run the following command (skip this if upgrading from a higher 2.0.n version):

   ```
   # ./mwe_setup.sh -update-registry
   ```

3. Run the upgrade script.

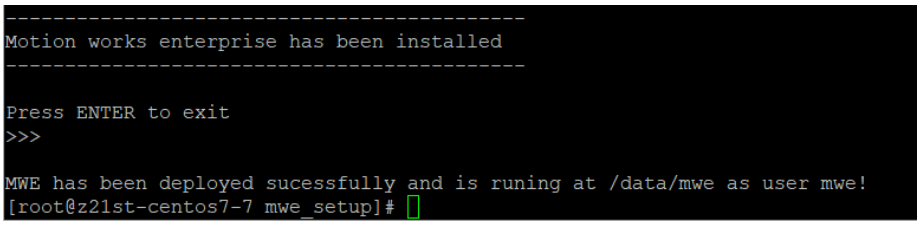
   - For an off-line upgrade, enter:

     ```
     # ./mwe_setup.sh --offline-upgrade
     ```

   - For an on-line upgrade, enter:

     ```
     # ./mwe_setup.sh --upgrade
     ```

4. When upgrade completes, press **Enter** to exit.

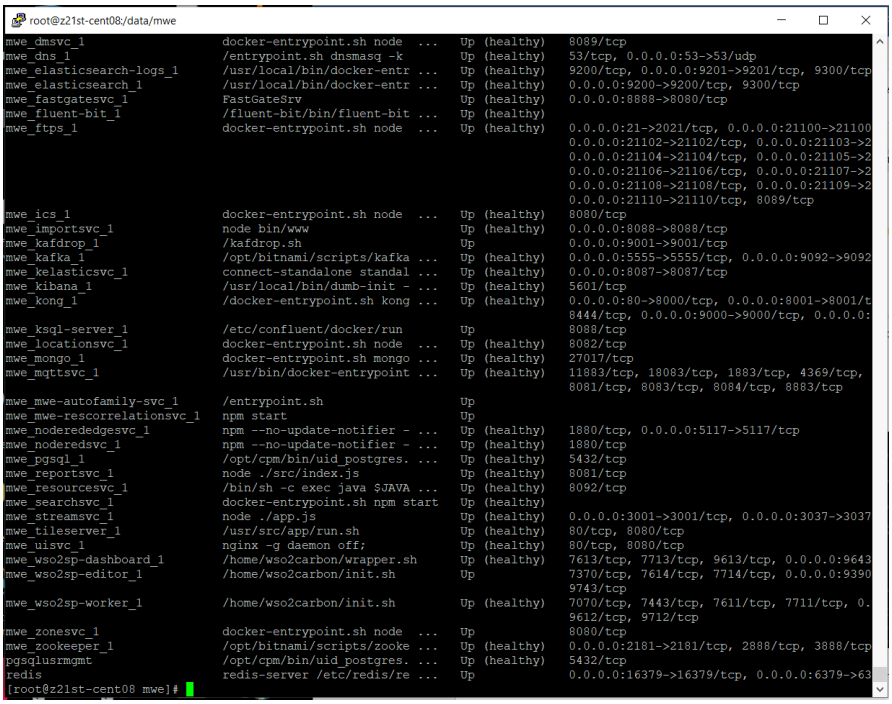


NOTE: When upgrading from 2.0.1/2.0.2 to a higher 2.0.m version, error messages may scroll in the Terminal/Putty window. Disregard these, as they do not affect installation. See Possible Error Messages During Upgrade from 2.01/2.02. If errors not included in this list appear, report them to Zebra Product Support.

Validating the Upgrade

Verify all MWE services are functional. Depending on server resources (CPU, memory), it may take a few minutes for all services to be operational. To check the status of the MWE services (Docker containers), run the following commands:

```
# cd /data/mwe
# docker-compose ps
```



Verify the **State** column displays **Up** for all containers. Some services also report a health condition and should show **healthy**.

NOTE: You may see several warnings if you run the docker-compose command under the root account. Disregard these. If you switch to the mwe account, the warnings do not display.

```
# su - mwe
# cd /data/mwe
# docker-compose ps
```

To switch back to the root account enter:

```
# su -
```

To restart all MWE services for any reason, either reboot the server or enter these commands:

```
# su - mwe (if prompted, the default password is Zebra123)
# cd /data/mwe
# ./mwe --restart
```

If your deployment includes a ZLA, upgrade the ZLA software to a version compatible with the MWE version. See Upgrading the ZLA.

Upgrading from MWE 2.0.5 to MWE 2.0.6

If upgrading from MWE 2.0.n where n<5 to MWE 2.0.6, first upgrade to 2.0.5 (see Upgrading from MWE 2.0.n to MWE 2.0.m), and then upgrade from 2.0.5 to 2.0.6 following the instructions in this section. You can perform an off-line upgrade or an on-line upgrade.

- Off-line upgrade does not require the MWE Linux server to have Internet access. Because the installation package is close to 8.5 GB, copying it prior to a scheduled upgrade is recommended.
- On-line upgrade requires that the MWE Linux server have open access to the Internet during installation.

CAUTION: If your MWE deployment includes passive RFID readers in Device Manager which are connected to MWE using either a non-secure connection or a secure connection with a self-signed SSL certificate for domain **zebramwe** (the default certificate installed with MWE), note that MWE 2.0.6 and later do not support non-secure connection with RFID readers, and passive RFID readers by default do not accept the **zebramwe** certificate. To upgrade from MWE 2.0.5 to 2.0.6 in this case, you must implement one of the following options immediately after upgrade, otherwise the readers cannot communicate with the MWE server:

- Install a valid CA SSL certificate on the MWE server.
- Install a self-signed certificate using the FQDN (Fully Qualified Domain Name) of the MWE server.
- Configure the MWE server so that readers accept the default **zebramwe** domain certificate.

Re-initialize the readers after implementing one of these options.

Refer to the *MWE Configuration Guide* for instructions on implementing one of the above options, and the *MWE Device Manager User Guide* for instructions on re-initializing readers.

CAUTION: If an external application uses the MWE API to pull or push data to the MWE server, note that MWE 2.0.6 enforces new strong password requirements for MWE local accounts, and rejects passwords that do not meet these requirements. To avoid disruptions in API functionality, coordinate with API users before upgrading to MWE 2.0.6 to ensure that account passwords used for API access are updated to comply with the new password policy. Refer to **Launching the MWE Web Client** in the *MWE 2.0 User Guide* for password requirements.

Take a Snapshot

Following best practices, capture a snapshot of your virtual machine before upgrading in case you need to restore the previous version.

Copying and Extracting the Installation Package

1. Using WinSCP, Putty, or a similar SSH client, copy the installation package to the **`/data`** directory on the Linux server:
   - For off-line upgrade, copy the **`mwe-containers-setup-2.0.m-offline.tar.gz`** installation package.
   - For on-line upgrade, copy the **`mwe-containers-setup-2.0.m-online.tar.gz`** installation package.
2. Using a Terminal window, Putty, or another SSH client, log into the MWE server using the root account (or sudo account with sufficient permissions).
3. Change the directory to **`/data`**.

   ```
   # cd /data
   ```
4. Extract the files.
   - For off-line upgrade, enter:

```
# tar -xvf  mwe-containers-setup-2.0.m-offline.tar.gz
```

- For on-line upgrade, enter:

```
# tar -xvf  mwe-containers-setup-2.0.m-online.tar.gz
```

NOTE: Occasionally characters are misinterpreted when copying a command from a document and pasting it into a Putty or Terminal window. Some commands may require manual entry.

Preparing the MWE Server for Upgrade

1. Still using the root account (or sudo account with sufficient permissions), change directory:

   ```
   # cd /data/mwe_setup
   ```

2. Run the MWE 2.0.5 to 2.0.6 migration script, which prepares the 2.0.5 server for upgrade to 2.0.6. This script changes, among other things, the permissions for many MWE files from the root account to the mwe account.

   ```
   # ./mwe-upgrade_to_2.0.6-migration.sh
   ```

3. At the prompt, enter '**y**' (yes) and press **Enter**.
4. Run the following command to extract the MWE files:

   ```
   # ./mwe_setup_admin.sh --extract-package
   ```

5. At the following prompt:

   `Is this server connected to the Internet? (y/n)`

   enter '**n**' for an offline installation, or '**y**' for an online installation.

Running the Upgrade Script

1. Switch to the **mwe** account (the default password is Zebra123).

   ```
   # su - mwe
   ```

2. Change the directory:

   ```
   $ cd /data/mwe
   ```

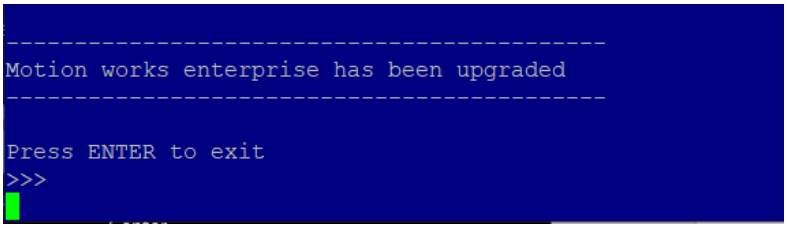
3. Run the upgrade script.
   - For off-line upgrade, enter:

     ```
     $ ./mwe --offline-upgrade
     ```

   - For on-line upgrade, enter:

     ```
     $ ./mwe --upgrade
     ```

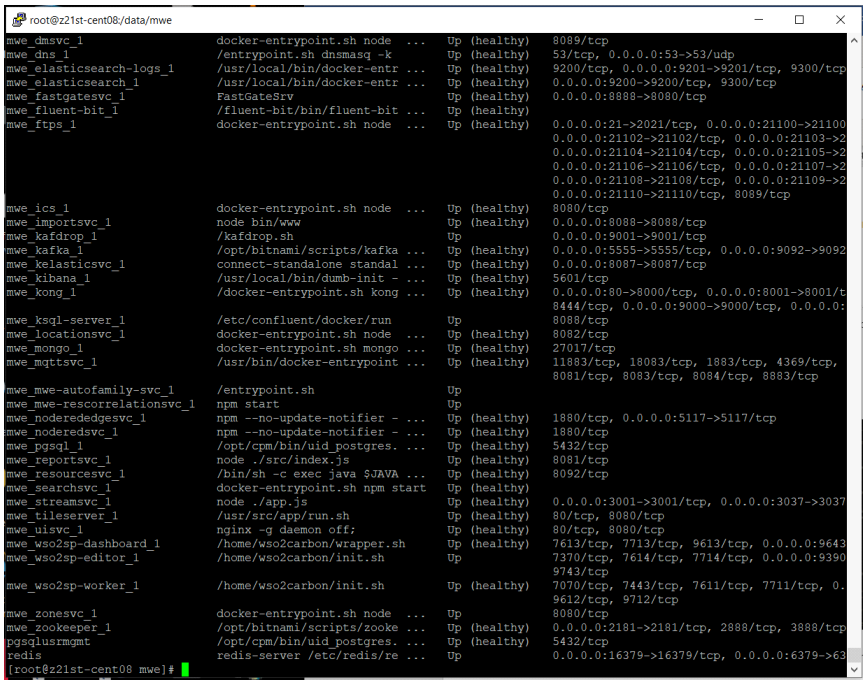
4. When upgrade completes, press **Enter** to exit.



Validating the Upgrade

Verify all MWE services are functional. Depending on the resources (CPU, memory) on your server, it may take a few minutes for all services to be up and running. To check the status of the MWE services (Docker containers), run the following commands:

```
$ cd /data/mwe
$ docker-compose ps
```



Verify that the **`State`** column displays **`Up`** for all containers. Some services also report a health condition and should show **`healthy`**.

NOTE: You may see several warnings if you run the **`docker-compose ps`** command under the root account. Disregard these. If you switch to the mwe account, the warnings do not appear.

```
# su - mwe
$ cd /data/mwe
$ docker-compose ps
```

To switch back to the root account enter:

```
# su -
```

To restart all MWE services for any reason, either reboot the server or use these commands:

```
# su - mwe (if prompted, the default password is Zebra123)
$ cd /data/mwe
$ ./mwe --restart
```

If your deployment includes a ZLA, upgrade the ZLA software to a version compatible with the MWE version. See Upgrading the ZLA.

Configuring Secure Connection for RFID Readers

This section when:

- Upgrading from MWE 2.0.5 to 2.0.6, and
- The deployment includes passive RFID readers in Device Manager, and
- The readers are connected to MWE using either a non-secure connection or a secure connection with a self-signed SSL certificate for domain **zebramwe** (the default certificate installed with MWE).

Refer to the top section of the **`/data/mwe/.env`** file on the MWE server to determine which type of connection is active:

- **`R2C_SECURE=false`** - non-secure connection
- **`R2C_SECURE=true and CERT_OPTION=3`** - **zebramwe** default certificate
- **`R2C_SECURE=true and CERT_OPTION=1`** - certificate provided by customer or IT
- **`R2C_SECURE=true and CERT_OPTION=2`** - self-signed certificate generated by MWE using the FQDN (Fully Qualified Domain Name) of the MWE server.

MWE 2.0.6 and later releases do not support non-secure connection with RFID readers, and passive RFID readers by default do not accept the **zebramwe** certificate. If the three conditions above apply to your deployment, you must implement one of the following options immediately after upgrade, otherwise the readers cannot communicate with the MWE server:

- Install a valid CA SSL certificate on the MWE server.
- Install a self-signed certificate using the FQDN (Fully Qualified Domain Name) of the MWE server.
- Configure the MWE server so that readers accept the default **zebramwe** domain certificate.

Re-initialize the readers after implementing one of these options.

Refer to the *MWE Configuration Guide* for instructions on implementing one of the above options, and the *MWE Device Manager User Guide* for instructions on re-initializing readers.

ZLA Software

This section includes information on installing, configuring, and upgrading the ZLA.

Installing ZLA Software

This section explains how to convert a Red Hat 8.x or Ubuntu 22 host into a ZLA (Zebra Location Appliance). See Minimum Server Specs for host requirements.

Red Hat 8.x Host

NOTE: Converting a Red Hat server to a ZLA requires installing some Red Hat libraries which cannot be distributed as part of an installer due to licensing restrictions. Therefore, during installation the server needs access to a Red Hat repository on the Internet or on the local network.

1. Connect a Putty/Terminal window to the Red Hat server. Login as root.
2. Install the required libraries:

   ```
   # yum install -y glibc.i686 java-1.8.0-openjdk libgcc.i686 libstdc++.i686
   nss-softokn-freebl.i686 nss-softokn-freebl.x86_64 vsftpd.x86_64 zip.x86_64 zlib.i686
   zlib.x86_64
   ```

3. For ZLA 2.0.5 or higher, install **`nodejs v.18`**:

   ```
   # dnf module install nodejs:18
   ```

4. Verify the firewalld daemon is running:

   ```
   # systemctl start firewalld
   # systemctl status firewalld
   ```

5. Using WinSCP or similar tool, copy the **`ZLA-2.0.n-m.i386.rpm`** package to the **`/home`** directory on the host server, and then run the following commands to install the ZLA software:

   ```
   # cd /home
   # rpm -ivh ZLA-2.0.n-m.i386.rpm --nodeps
   ```

NOTE: If an error displays regarding the icsagent service failing to start, this is expected. The icsagent starts successfully when a configuration file is published. See Configuring the ZLA.

6. Verify the installed ZLA software version:

   ```
   # rpm -qa | grep ZLA
   ```

Verify the **`ZLA-2.0.n-m.i386.rpm`** package you are installing is compatible with the MWE version running on the MWE server to which the ZLA will connect.

Ubuntu 22 Host

The Ubuntu host requires access to the Internet or a local Ubuntu repository.

1. Connect a Putty/Terminal window to the Ubuntu server. Switch to the root account.
2. Install the required libraries:

```
# apt update
# apt upgrade
# apt-get install lib32z1 lib32stdc++6
# apt-get install nodejs
# apt-get install zip
# apt-get install openjdk-11-jre-headless
# apt-get install vsftpd
# apt-get install net-tools
# apt-get install firewalld
# apt-get -f --reinstall install python3-problem-report
```

3. Edit the **`/etc/ssl/openssl.cnf`** file and comment out the following line:

```
providers = provider_sect
```

   so that it becomes:

```
# providers = provider_sect
```

4. Using WinSCP or similar tool, copy the **`ZLA-2.0.n-m.i386.rpm`** package to the **`/home/mwuser`** directory on the host server, and then run the following commands to install the ZLA software:

```
# dpkg --add-architecture i386
# dpkg -i ZLA-2.0.n-m.i386.deb
```

5. Enable services:

```
# systemctl enable icsagent
# systemctl enable zls
# systemctl start zls
# systemctl start icsagent
```

NOTE: The status of the icsagent and zls services displays **`Activating`** or **`Stopped`**. This is expected. The status displays **`Running`** once the ZLA is registered and a site configuration is published. Refer to the *MWE 2.0 Configuration Guide* for details.

6. Verify the version of the installed ZLA software:

```
# dpkg-query -l | grep zla
```

   Ensure the **`ZLA-2.0.n-m.i386.rpm`** package being installed is compatible with the MWE version running on the MWE server to which the ZLA will connect.

Ubuntu 22 OVF Template

Alternatively, Zebra also offers an Ubuntu 22 OVG template to load in the hypervisor environment that includes all required libraries and ZLA software already installed. However, upgrading the ZLA software may be required to make it compatible with the MWE version on the MWE server to which the ZLA will connect. See Upgrading the ZLA for instructions.

Configuring the ZLA

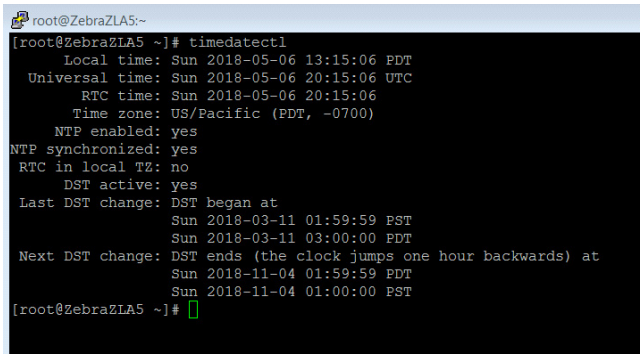
This section provides instructions on setting the IP address, time zone, system time, hostname, and more on a ZLA.

Network Configuration

A static IP address must be assigned to the ZLA, and the ZLA must have network connectivity to the MWE server and to the sensors or readers that it will communicate with. See Network Connections and Ports for a diagram of the required ports and network connections.

ZLA Time and Date

1. Log into the ZLA using the root account (obtain login credentials from Zebra) and open a Terminal window using Putty or a similar SSH client to remotely access the ZLA.
2. To check the current time and date on the ZLA, use the **`timedatectl`** command:



3. To set a US time zone (US/Pacific (PDT) in the example above), use one of the following commands:

```
# timedatectl set-timezone US/Pacific
# timedatectl set-timezone US/Mountain
# timedatectl set-timezone US/Central
# timedatectl set-timezone US/Eastern
```

   To see a list of worldwide Time Zones, use the command:

```
# timedatectl list-timezones
```

4. The ZLA includes an NTP client (chrony) that automatically syncs the system clock with NTP servers on the Internet. If the ZLA does not have access to the Internet, use an NTP server on your local network to set the time by adding the line highlighted in red to the **`/etc/chrony.conf`** file, replacing the IP address with the address or name of your local NTP server.

```
# Use public servers from the pool.ntp.org project.
# Please consider joining the pool (http://www.pool.ntp.org/join.html).
server 0.centos.pool.ntp.org iburst
server 1.centos.pool.ntp.org iburst
server 2.centos.pool.ntp.org iburst
server 3.centos.pool.ntp.org iburst
server 192.168.1.50 prefer iburst
```

5. Restart the NTP client to apply the change:

```
# systemctl restart chronyd
```

6. If necessary, to manually set the date and time, use the following commands:

   ```
   # timedatectl set-ntp 0
   ```

   (disables the NTP client)

   ```
   # timedatectl set-time YYYY-MM-DD
   ```

   (for example, **`timedatectl set-time 2018-05-07`**)

   ```
   # timedatectl set-time HH:MM:SS
   ```

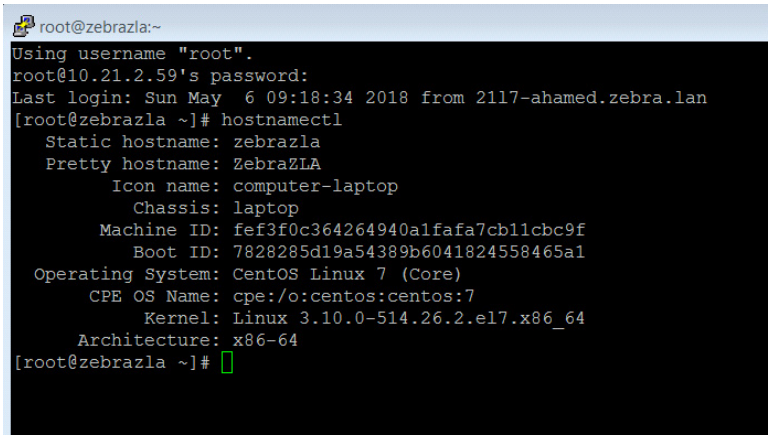
   (for example, **`timedatectl set-time 14:45:05`**)

NOTE: A ZLA supports only one time zone. Therefore, all site maps associated with a ZLA in MWE use the same time zone. Use separate ZLAs for sites in different time zones.

Changing the Hostname

Log into the ZLA using the root account (obtain login credentials from Zebra) and open a Terminal window. Use Putty or a similar SSH client to remotely access the ZLA.

A ZLA ships with the default hostname **`ZebraZLA`**. Changing this hostname is recommended. The following command line prompt shows this default name. The **`hostnamectl`** command provides more details:



To change the hostname to **`NewZLAName`** use the command:

```
# hostnamectl set-hostname NewZLAName
```

This command changes the hostname at the Kernel level (static name). The name displayed at the command prompt (transient name) is updated after a reboot. Issue a reboot command to reboot the ZLA.

WHERENET_HOST_IP

If the ZLA network deploys WhereLAN or DVR sensors and has more than one network interface connected, you must specify to the location engine running on the ZLA which interface is receiving data from the location sensors. To do this, set the WHERENET_HOST_IP environment variable to the static IP address of the network card on the same subnet as the location sensors.

1. Connect to the ZLA using Putty, WinSCP, or similar SSH client.
2. If using a terminal window, change the directory:

   ```
   # cd /etc/systemd/system
   ```

   If using WinScp, browse to this folder.
3. Open the file **`/etc/system/system/zls.service`** for editing using a text editing command such as **vi**, or double-click the file if using WinSCP. File contents are as follows:

   ```
   [Unit]
   Description=Zebra Location Service
   After=network.target

   [Service]
   Type=forking
   Environment=ZEBRA_HOME=/opt/zebra
   Environment=LD_LIBRARY_PATH=/opt/zebra/zla/zlpcore/bin
   Environment=WHERENET_HOST_IP=192.168.30.120
   ExecStart=/opt/zebra/zla/zlpcore/bin/start-zls.sh
   ExecStop=/opt/zebra/zla/zlpcore/bin/stop-zls.sh
   KillMode=none
   Restart=always
   RestartSec=15
   StartLimitInterval=60
   StartLimitBurst=3

   [Install]
   WantedBy=multi-user.target
   ```
4. Add or edit the line highlighted in red above, setting the IP address to the static IP of the network interface on the same subnet as the location sensors.
5. Save and close the file.
6. Run the following command in a terminal window to apply the changes:

   ```
   # systemctl daemon-reload
   ```

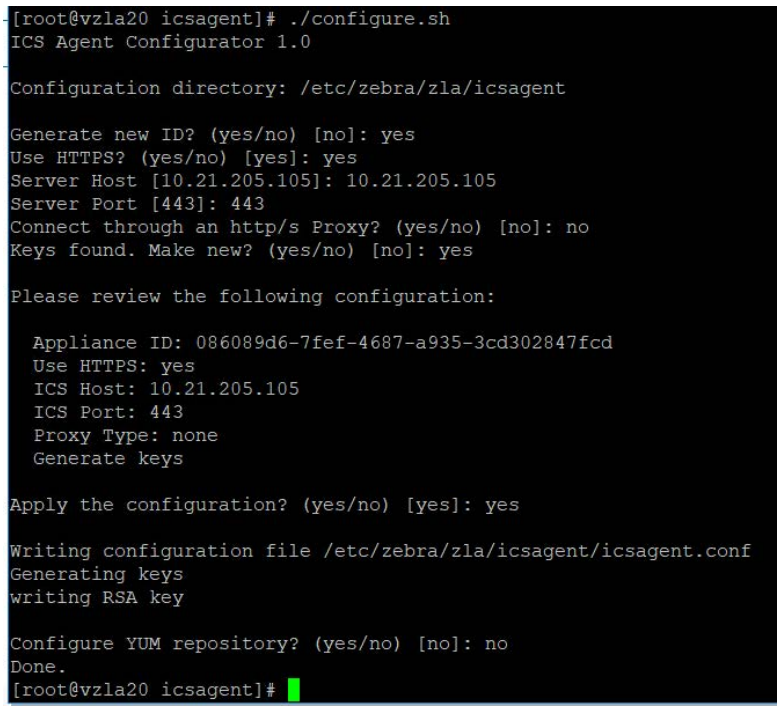
Registering the ZLA

The ZLA appliance must be registered with the MWE server to allow the ZLA to forward data to the server, and the MWE web client to monitor, configure, and update the ZLA. To register the ZLA with the server:

1. Using Putty or a similar SSH client, log into the ZLA using the root account (obtain login credentials from Zebra).
2. Change the directory and run the configure script:

   ```
   # cd /opt/zebra/zla/icsagent
   # ./configure.sh
   ```

3. Respond to prompts as follows. Press **Enter** to accept the values in square brackets [ ] (the current or default value). For **Server Host**, enter the IP address or fully qualified domain server name of the MWE Linux server.



   If there is a proxy server between the ZLA and the MWE server, answer **yes** to the server proxy prompt and enter the URL for the proxy server.

4. Run the register script:

   ```
   #  ./register.sh
   ```

5. When prompted, enter the Username / Password (default is **admin** / **admin**) and enter a name for the ZLA. This name is displayed in the MWE web client. Using the same name previously configured as hostname is recommended.

```
[root@vzla20 icsagent]# ./register.sh
ICS Agent Registration Utility

Reading configuration from /etc/zebra/zla/icsagent/icsagent.conf
MotionWorks server is at 10.21.205.45:443

Please enter your server access credentials (Ctrl-C to exit)
  Username: admin
  Password: *****

Enter the appliance name: vzla20

Registering appliance...
Done.
[root@vzla20 icsagent]#
```

6. Restart the icsagent daemon:

```
# systemctl restart icsagent
```

Upgrading the ZLA

This section explains how to upgrade ZLA software hosted on CentOS 7.9, Red Hat 7.9, Red Hat 8.x, and Ubuntu 22 servers.

Centos and Red Hat Hosts

1. Using WinSCP or similar tool, copy the **`ZLA-2.0.n-m.i386.rpm`** package to the **`/home`** directory on the ZLA. Using Putty or similar SSH client, log into the ZLA using the root account.
2. Verify that the firewalld daemon status is reported as **`active (running)`**:

```
# systemctl start firewalld
# systemctl status firewalld
```

3. Verify the current version of the ZLA software before upgrading:

```
# rpm -qa | grep ZLA
```

4. If upgrading to ZLA 2.0.5 or higher, use the **`node -v`** command to determine if the ZLA host has the correct nodejs version installed as indicated in the following table.

ZLA Operating System	Required nodejs Version
CentOS 7.9	17.9.0 or 17.9.1
Red Hat 7.9	17.9.0 or 17.9.1
Red Hat 8.x	18.x.x

If the host does not have the required version, upgrade the nodejs:

a. Run the following command(s) to remove the installed version of nodejs and the ZLA software from the host server. This does not remove the configuration files, so there is no need to re-configure or re-register the ZLA.

CentOS 7.9 or Red Hat 7.9:

```
# curl -sL https://rpm.nodesource.com/setup_17.x | bash -
```

```
#yum remove -y nodejs
```

Red Hat 8.x:

```
# yum remove -y nodejs
```

b. Run the following command to install the new ZLA software.

CentOS 7.9 or Red Hat 7.9:

```
# sudo yum install -y nodejs
```

Red Hat 8.x:

```
# dnf module install nodejs:18
```

5. If you upgraded nodejs on your ZLA, run the following commands to install the new **`ZLA-2.0.n-m.i386.rpm`** package:

```
# cd /home
# rpm -ivh ZLA-2.0.n-m.i386.rpm --nodeps
```

If you did not need to upgrade nodejs on your ZLA, run the following commands to upgrade to the new **`ZLA-2.0.n-m.i386.rpm`** package:

```
# cd /home
# rpm -Uvh ZLA-2.0.n-m.i386.rpm
```

6. Verify the version of the newly installed ZLA software:

```
# rpm -qa | grep ZLA
```

7. Restart services:

```
# systemctl restart icsagent
# systemctl restart zls
```

8. Check services status:

```
# systemctl status icsagent
# systemctl status zls
```

Verify the icsagent status is **`active (running)`** if the ZLA was previously configured by running the **`./configure.sh`** command (see Registering the ZLA). Otherwise, the status displays **`failed`** or **`activating`**; this is expected.

For the zls service, the status may show **`activating`** or **`failed`** if no **`site.json`** configuration file is published to the ZLA. The status becomes **`Running`** once a valid **`site.json`** configuration file is published with System Builder. Refer to the *MWE Configuration Guide*.

Ubuntu 22 Host

1. Using WinSCP or similar tool, copy the **ZLA-2.0.n-m.i386.rpm** package to the **/home/mwuser** directory on the ZLA host.
2. Connect a Putty/Terminal window to the ZLA. Switch to the root account.
3. Determine the version of the installed ZLA software:

   ```
   # dpkg-query -l | grep zla
   ```

4. Run the following command to upgrade the ZLA software:

   ```
   # dpkg -i ZLA-2.0.n-m.i386.deb
   ```

5. Restart services:

   ```
   # systemctl restart zls
   # systemctl restart icsagent
   ```

NOTE: The status of the icsagent and zls services displays **activating** or **stopped** if the ZLA is not registered with MWE and a site configuration file was not published with System Builder. The status displays **running** once the ZLA is registered and a site configuration is published. Refer to the *MWE 2.0 Configuration Guide*.

6. Verify the version of the installed ZLA software:

   ```
   # dpkg-query -l | grep zla
   ```

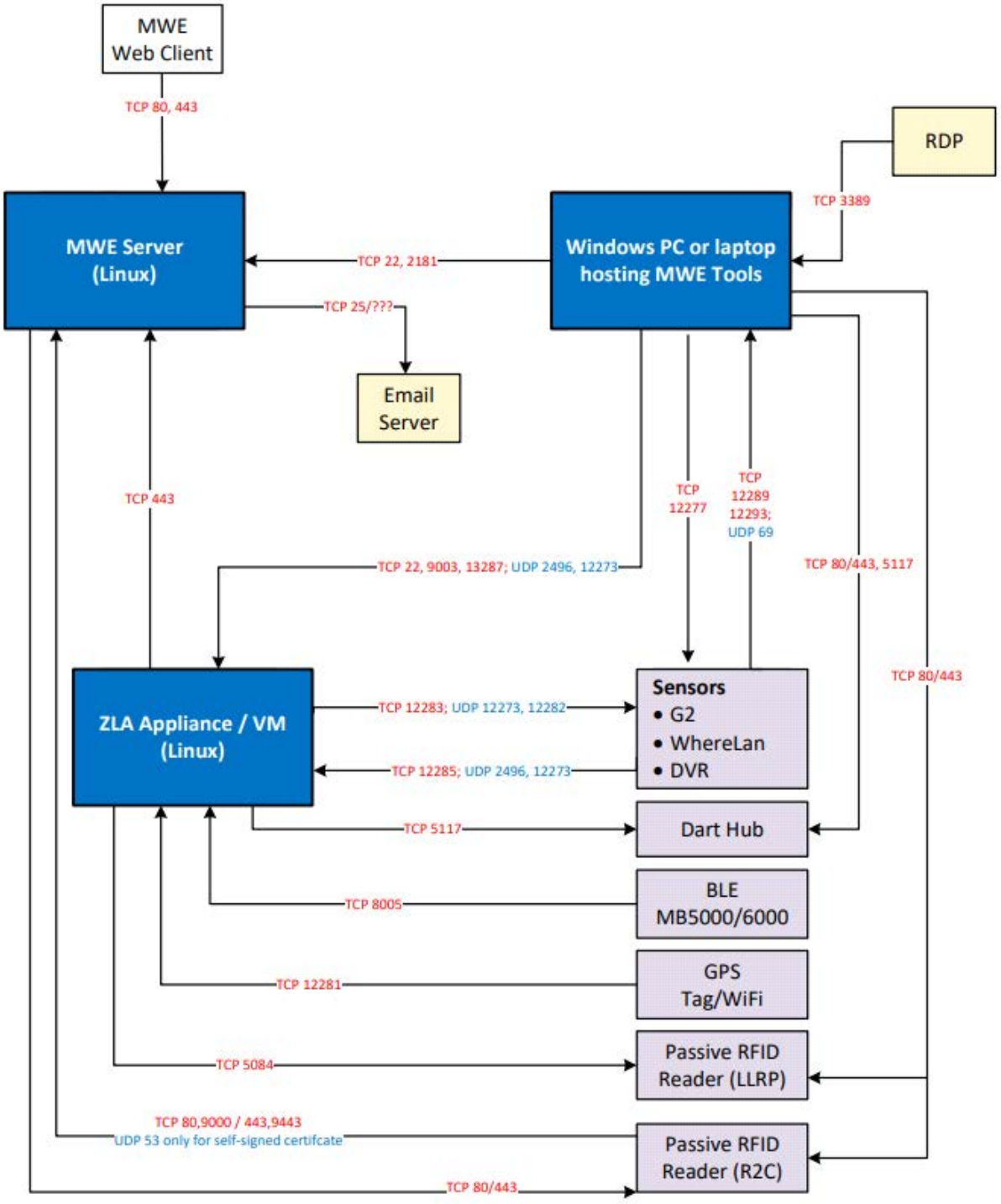
   Verify that the **ZLA-2.0.n-m.i386.rpm** package being installed is compatible with the MWE version running on the MWE server that the ZLA will connect to.

Network Connections and Ports

This section provides an MWE network block diagram and details network ports.

MWE Network Block Diagram

The following diagram shows ports that must be open in an MWE 2.0.x deployment, depending on the type of sensor/reader used.



Ports

Under normal operation, sensors/readers communicate with a ZLA or directly with the MWE server, and the ZLA communicates with the MWE server. This requires open ports and traffic per the MWE Network Block Diagram. The required ports depend on the type of sensor/reader deployed.

The following tables provide details on each port.

Table 1 MWE Server Ports

Source	Destination	Protocol & Port Number	Description
Passive RFID reader running firmware 3.9.16	MWE server	TCP 21, TCP 21100-21110	Required by passive RFID readers for downloading an SSL certificate when: • Reader is running firmware version 3.9.16 • Reader is added to MWE via Device Manager • MWE is configured to use secure connection (https) to reader Note: RFID reader firmware 3.10.30 and higher uses https and port 443 to download the certificate.
Passive RFID readers	MWE server	TCP 80 or 443	Used by passive RFID readers added to MWE via Device Manager to post tag blink data. If MWE is configured to use a secure connection (https) with the readers, 443 is used. For non-secure connection, port 80 is used.
Passive RFID readers	MWE server	UDP 53	Required by passive RFID readers added to MWE via Device Manager when MWE is configured to use a secure connection using a self-signed certificate with domain name **zebramwe**.
Passive RFID readers	MWE server	TCP 9000 or 9443	Required by passive RFID readers added to MWE via Device Manager, and used for management commands and health and status reporting. If MWE is configured to use a secure connection (https) with the readers, port 9443 is used. For non-secure connection, port 9000 is used.
Browser – MWE web client	MWE server	TCP 80 or 443	Between web client and MWE server. Used for http/s client connections.
ZLA	MWE server	TCP 80 or 443	Used by ZLA for http/s and web-socket connection to MWE server.
Kafka Tool, Offset Explorer	MWE server	TCP 2181	For connecting Kafka tool (Offset Explorer) to the MWE server for troubleshooting purposes.
Tool on PC	MWE server	TCP 9092, 9093	For connecting debugging tool to Kafka for troubleshooting purposes.
SSH Client	MWE server	TCP 22	Used by SSH client to connect to MWE server.
MWE server	Zebra yum repo web site	TCP 5000	Between MWE server and Zebra yum repo web site. Required only during on-line installation.

Table 2 ZLA Ports

Source	Destination	Protocol & Port Number	Description
SSH client	ZLA	TCP 22	Used by SSH clients to remotely connect to the ZLA.
WhereLAN III and DVR sensors	ZLA	UDP 2496, UDP 12273, TCP 12285	Communication between sensors and ZLA.
Telnet session or 3rd party app	ZLA	TCP 9003	Locate data stream for third party applications. Also used for diagnostics and troubleshooting.
BLE Receivers	ZLA	TCP 8005	Default port used by MPACT BLE receivers to send data to a ZLA. This is configurable.
MWE tools on Windows PC	ZLA	TCP 13287	Used by ZLA diagnostic/troubleshooting tools installed on Windows PC.
GPS tag	ZLA	TCP 12281	Used by GPS tags sending data to a ZLA via a WiFi access point.

Table 3 Sensors, Readers, and Dart Hubs Ports

Source	Destination	Protocol & Port Number	Description
ZLA	WhereLAN III and DVR sensors	UDP 12273, UDP 12282, TCP 12283	Communication between sensors and ZLA.
MWE Tools on Windows PC	WhereLAN III and DVR	TCP 12277	Used by diagnostic/troubleshooting tool installed on Windows PC.
ZLA	Passive RFID reader	TCP 5084	ZLA connects to this port on a passive RFID reader using LLRP protocol.
MWE server	Passive RFID reader	TCP 80 or 443	Used by MWE server to connect to RFID reader when running R2C application.
ZLA	Dart Hub	TCP 22	Used by ZLA to subscribe to Dart hub using SSH connection.
Telnet session on Windows PC	Dart Hub	TCP 5117	For monitoring data output on Dart hub via Telnet (port must be enabled/open on Dart hub).
Locate Analyzer tools on Windows PC	Dart Hub	TCP 5111, 5115, 5116, 5117	Used by Locate Analyzer tool to collect raw data from a Dart hub.
MWE server, web browser	Dart Hub	TCP 80 or 443	Used by web client and MWE server.

Table 4 Ports on Windows PC Running MWE Tools

Source	Destination	Protocol & Port Number	Description
WhereLAN III and DVR sensors	Windows PC running MWE Tools	TCP 12289, TCP 12293, UDP 69	Communication between sensors and Sensor Analyzer tool.

Firmware Versions

MWE supports a variety of readers and locating sensors:

- Passive RFID readers
- BLE beacons and receivers
- WhereLAN sensors (ISO 24730)
- Dart UWB sensors (ISO 24730-61)
- DVR UWB sensors
- GPS tags

The firmware version on a reader or sensor must be compatible with the MWE 2.0.x version. This section provides information on firmware compatibility, but consult Zebra Support for the latest available or recommended firmware versions.

WhereLAN and DVR Sensors

WhereLAN sensors require firmware v.6.5.2 or later. Version 6.5.9 is recommended for sites that use WhereLAN sensors to locate WhereNet tags.

DVR sensors require firmware v.5.2.1 or later.

Use MWE's Sensor Analyzer tool to perform firmware upgrades for these sensors, described in separate documentation.

DART Hub

DART UWB sensors connect to a Dart Hub, which in turn forwards location information about DART tags to MWE. The DART hub must be running software version 5.6.1 or later.

BLE Beacons and Receivers

Consult Zebra Support.

Passive RFID Readers

The following tables show version combinations tested by Zebra. Recent firmware versions (such as 3.26.90 and 3.28.1) may operate properly with previous MWE versions (such as 2.0.3 or 2.0.4), but following these tables is recommended.

Reader Models: Zebra FX7500, FX9600

MWE Version	Firmware Version	Connection Type to MWE		
		Non-secure	Secure with Self-signed SSL Certificate	Secure with CA SSL Certificate
MWE 1.4	3.9.16	Yes	Yes	Yes
MWE 2.0.2	3.10.30	Yes	Yes	Yes
MWE 2.0.3	3.21.23 3.24.43 3.24.48	Yes	Yes	MWE patch required for 3.24.48
MWE 2.0.4	3.21.23 3.24.43 3.24.48 3.25.70	Yes	Yes	MWE patch required for 3.24.48, 3.24.70
MWE 2.0.5	3.21.23 3.24.43 3.24.48 3.25.70 3.26.90	Yes	Yes	Yes
MWE 2.0.6	3.21.23 3.24.43 3.24.48 3.25.70 3.26.90 3.28.1	No	Yes	Yes

Reader Model: Zebra FXR90

MWE Version	Firmware Version	Connection Type to MWE		
		Non-secure	Secure with Self-signed SSL Certificate	Secure with CA SSL Certificate
MWE 2.0.2 - 2.0.5	N/A	N/A	N/A	N/A
MWE 2.0.6	2.0.10	No	Yes	Yes

Reader Model: ATR

MWE Version	Firmware Version	Connection Type to MWE		
		Non-secure	Secure with Self-signed SSL Certificate	Secure with CA SSL Certificate
MWE 2.0.3	3.24.48	Yes	Yes	MWE patch required
MWE 2.0.4	3.24.48 3.25.70	Yes	Yes	MWE patch required
MWE 2.0.5	3.24.48 3.25.70 3.26.90	Yes	Yes	Yes
MWE 2.0.6	3.24.48 3.25.70 3.26.90	No	Yes	Yes

Possible Error Messages During Upgrade from 2.01/2.02

When upgrading from 2.0.1/2.0.2 to a higher 2.0.x, the following error messages may scroll in the Terminal/Putty window. Disregard these, as they do not affect installation. If other errors not included in the list display, report them to Zebra Product Support.

ERROR: could not open extension control file "/usr/local/share/postgresql/extension/pgaudit.control": No such file or directory

ERROR: extension "pgaudit" does not exist

ERROR: must be owner of extension pgcrypto

ERROR: permission denied to create extension "pg_stat_statements"

ERROR: extension "pg_stat_statements" does not exist

ERROR: could not open extension control file "/usr/local/share/postgresql/extension/pgaudit.control": No such file or directory

ERROR: extension "pgaudit" does not exist

ERROR: must be owner of extension pgcrypto

Error: No such container: mwe_wso2sp-worker_1

ERROR: Failed to check mongo version: MongoDB not running. Will attempt to upgrade from 4.0 to 4.4.

error while removing network: network mwe_default id a4157ef19739f5babeaad83b07f705f825fcb4796269689388a3030dd52e7382 has active endpoints

Error response from daemon: Container 2c8707588725aa5a553e8c4fc54fbb6e3bc3436b172d5c6dfc833ab21cf2bfee is not running

error while removing network: network mwe_default id a4157ef19739f5babeaad83b07f705f825fcb4796269689388a3030dd52e7382 has active endpoints

Error response from daemon: No such container: temporal_postgres

Error: No such container: temporal_postgres