**DOCUMENTATION FOR END-NODE/SENSOR NODE AND GATEWAY:**

**CHIRPSTACK SETUP INSTRUCTIONS:**

# General install:

The ChirpStack LoRaWAN® Network Server components can be setup in multiple ways. For all cases, downloads can be found at the following URLs:

* [ChirpStack Gateway Bridge downloads](https://www.chirpstack.io/gateway-bridge/downloads/)
* [ChirpStack Network Server downloads](https://www.chirpstack.io/network-server/downloads/)
* [ChirpStack App Server downloads](https://www.chirpstack.io/application-server/downloads/)

## Manual

In this case you need to download the pre-compiled binaries for each component and setup scripts so that these components will be started on boot (if desired).

For this you could use [systemd](https://en.wikipedia.org/wiki/Systemd) unit-files or [init](https://en.wikipedia.org/wiki/Init) script, based on the used Linux distribution.

## Debian / Ubuntu repository

ChirpStack provides Debian / Ubuntu .deb packages which can be downloaded from the ChirpStack Deb repository. To guarantee compatibility, for each major ChirpStack version a separate repository is provided. Please refer to the [Debian / Ubuntu install guide](https://www.chirpstack.io/project/guides/debian-ubuntu/) for a step-by-step guide.

## Docker

ChirpStack also provides Docker images which for example can be used with [Docker Compose](https://docs.docker.com/compose/). See the [Docker install](https://www.chirpstack.io/project/install/docker/) for more information.

**REQUIREMENTS:**

Before getting started with the ChirpStack LoRaWAN® Network Server components, there are a couple of requirements that needs to be satisfied.

## MQTT broker

ChirpStack makes by default use of MQTT for publishing and receiving application payloads. [Mosquitto](http://mosquitto.org/) is a popular open-source MQTT server, but any MQTT broker implementing MQTT 3.1.1 should work. In case you install Mosquitto, make sure you install a recent version.

MQTT is used by ChirpStack Gateway Bridge, ChirpStack Network Server, and ChirpStack Application Server.

### Install

#### Debian / Ubuntu

To install Mosquitto:

sudo apt install mosquitto

#### Other platforms

Please refer to the [Mosquitto download](https://mosquitto.org/download/) page for information about how to setup Mosquitto for your platform.

## PostgreSQL database

The ChirpStack components are using [PostgreSQL](https://www.postgresql.org/) for persistent data-storage. Note that PostgreSQL 9.5+ is required and that each component requires its own database to avoid schema conflicts. When running multiple ChirpStack Network Server instances to support multiple LoRaWAN® regions, you must create a database for each region!

There is no need to run multiple PostgreSQL instances as a single instance can host multiple databases.

PostgreSQL is used by ChirpStack Network Server and ChirpStack Application Server.

### Install

#### Debian / Ubuntu

To install the PostgreSQL:

sudo apt install postgresql

#### Other platforms

Please refer to the [PostgreSQL download](https://www.postgresql.org/download/) page for information how to setup PostgreSQL on your platform.

## Redis database

The ChirpStack components are storing all non-persistent data into a [Redis](http://redis.io/) datastore. Note that at least Redis 2.6.0 is required.

Redis is used by ChirpStack Network Server and ChirpStack Application Server.

### Install

#### Debian / Ubuntu

To Install Redis:

sudo apt install redis-server

# Docker install

ChirpStack provides [Docker](https://www.docker.com/) images for all project components. An overview of available images can be found at: <https://hub.docker.com/u/chirpstack/>. Please refer to [Quickstart Docker Compose](https://www.chirpstack.io/project/guides/docker-compose/) for a guide on getting started with Docker Compose.

# Quickstart Docker Compose

[Docker Compose](https://docs.docker.com/compose/) (part of Docker) makes it possible to orchestrate the configuration of multiple Docker containers at once using a docker-compose.yml file.

## Requirements

### Install Docker

Please refer to the [Get Started with Docker](https://www.docker.com/get-started) guide to install Docker for MacOS or Windows. When installing Docker on Linux, please refer to one of the following guides:

* [CentOS](https://docs.docker.com/install/linux/docker-ce/centos/" \l "install-docker-ce)
* [Debian](https://docs.docker.com/install/linux/docker-ce/debian/)
* [Fedora](https://docs.docker.com/install/linux/docker-ce/fedora/)
* [Ubuntu](https://docs.docker.com/install/linux/docker-ce/ubuntu/)

### Install Compose

To install Docker Compose on Linux, please refer to the [Install Compose on Linux systems](https://docs.docker.com/compose/install/" \l "install-compose) guide. You can skip this step for MacOS and Windows.

## ChirpStack stack

### Configure

ChirpStack provides an example docker-compose.yml file that you can use as a starting-point. This example can be found at <https://github.com/brocaar/chirpstack-docker> and also contains more documentation.

To clone this repository, you need to execute the following commands:

git clone https://github.com/brocaar/chirpstack-docker.git

cd chirpstack-docker

### Start

After you have updated the configuration, you can run the following command to start all Docker containers:

docker-compose up

Please note that the first time you execute this command, there might be some errors logged as the database needs to be initialized.

## Add Network Server

As each container has its own hostname, you must use the hostname of the networkserver container when adding the network-server in the ChirpStack Application Server web-interface.

When using the above example, it means that you must enter chirpstack-network-server:8000 as the network-server hostname:IP. See [Network Servers](https://www.chirpstack.io/application-server/use/network-servers/) for more information.

To use Chrpstack on Dragino, we follow the instructions given to us at the link: https://www.chirpstack.io/gateway-bridge/gateway/dragino/

### Configure Packet Forwarder

In the Dragino LG-308 web-interface, you need to configure the Packet Forwarder so that it forwards its data to localhost on port 1700.

* In the Service menu, click on LoRaWAN Gateway
* Make sure the following settings are set:
* IoT Service: \_LoRaWAN / RAW Forwarder
* Service Provider: --custom--
* LoRaWAN Server Address: localhost or the remote\_server\_address
* Server port for upstream: 1700
* Server port for downstream: 1700

Click Save & Apply.

### Install ChirpStack Gateway Bridge

#### SSH into the gateway

The first step is to login into the gateway using ssh:

ssh root@GATEWAY-IP-ADDRESS

The default password is dragino.

#### Download IPK

Find the latest package at https://artifacts.chirpstack.io/vendor/dragino/LG308/ and copy the URL to your clipboard. Then on the gateway use wget to download the IPK package. It is important you download the package to /tmp. Example for chirpstack-gateway-bridge\_3.9.2-r1\_mips\_24kc.ipk:

cd /tmp

wget https://artifacts.chirpstack.io/vendor/dragino/LG308/chirpstack-gateway-bridge\_3.9.2-r1\_mips\_24kc.ipk

#### Install IPK

Use the opkg package-manager to install the downloaded package. Example:

opkg install chirpstack-gateway-bridge\_3.9.2-r1\_mips\_24kc.ipk

Note: In case of an upgrade, it is recommeded to first uninstall the chirpstack-gateway-bridge package using opkg remove ... and then install the new version using opkg install .... Configuration files will be maintained.

#### Configuration

To connect the ChirpStack Gateway Bridge with your MQTT broker, you must update the ChirpStack Gateway Bridge configuration file, which is located at: /etc/chirpstack-gateway-bridge/chirpstack-gateway-bridge.toml.

#### (Re)start and stop commands

Use the following commands to (re)start and stop the ChirpStack Gateway Bridge Service:

# start

/etc/init.d/chirpstack-gateway-bridge start

# stop

/etc/init.d/chirpstack-gateway-bridge stop

# restart

/etc/init.d/chirpstack-gateway-bridge restart

**udo udo udo udo Docker Install on AWS EC2:**

On an AWS Ubuntu AMI, the commands to get Docker are located at url: <https://docs.docker.com/engine/install/ubuntu/>

These commands are:

sudo apt-get update

$ sudo apt-get install \

apt-transport-https \

ca-certificates \

curl \

gnupg-agent \

software-properties-common\

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -

Verify that you now have the key with the fingerprint 9DC8 5822 9FC7 DD38 854A E2D8 8D81 803C 0EBF CD88, by searching for the last 8 characters of the fingerprint.

The command to verify this fingerprint is:

***sudo apt-key fingerprint 0EBFCD88***

$ sudo add-apt-repository \

"deb [arch=amd64] https://download.docker.com/linux/ubuntu \

$(lsb\_release -cs) \

stable"

Install Docker Engine

1. Update the apt package index, and install the latest version of Docker Engine and containerd, or go to the next step to install a specific version:

$ sudo apt-get update

$ sudo apt-get install docker-ce docker-ce-cli containerd.io

Verify that Docker Engine is installed correctly by running the hello-world image.

$ sudo docker run hello-world

To install docker-compose:

sudo curl -L "https://github.com/docker/compose/releases/download/1.27.0/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose

sudo chmod +x /usr/local/bin/docker-compose