

LANSITEC[®]

INSTALL CHIRPSTACK ON CENTOS

Doc. No. 930-001322

REV: 1.0.1 2020/12/22

Prerequisites

OS requirements:

To install Docker Engine, you need a maintained version of CentOS 7 or 8. Archived versions aren't supported or tested.

The centos-extras repository must be enabled. This repository is enabled by default, but if you have disabled it, you need to re-enable it.

1. Docker Community Engine

Before you install Docker Engine for the first time on a new host machine, you need to set up the Docker repository. Afterwards, you can install and update Docker from the repository.

1.1 Set up the repository

Install the `yum-utils` package (which provides the `yum-config-manager` utility) and set up the stable repository.

```
$ sudo yum install -y yum-utils \    device-mapper-persistent-data \    lvm2
$ sudo yum-config-manager \        --add-repo \
```

```
https://download.docker.com/linux/centos/docker-ce.repo
```

1.2 Install Docker Engine

Install the *latest version* of Docker Engine and containerd, or go to the next step to install a specific version:

```
$ sudo yum install docker-ce docker-ce-cli containerd.io
```

Start Docker.

```
$ sudo systemctl start docker
```

1.3 Verify

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

```
$ sudo docker run hello-world
```

1.4 Upgrade Docker Engine

To upgrade Docker Engine, download the newer package file and repeat the 1.1 installation procedure, using `yum -y upgrade` instead of `yum -y install`, and pointing to the new file.

1.5 Uninstall Docker Engine

1.5.1 Uninstall the Docker Engine, CLI, and Containerd packages:

```
$ sudo yum remove docker-ce docker-ce-cli containerd.io
```

1.5.2 Images, containers, volumes, or customized configuration files on your host are not automatically removed. To delete all images, containers, and volumes:

```
$ sudo yum remove docker-ce docker-ce-cli containerd.io
```

You must delete any edited configuration files manually.

2. Docker Compose

Docker Compose relies on Docker Engine for any meaningful work, so make sure you have Docker Engine installed either locally or remote, depending on your setup.

On Linux, you can download the Docker Compose binary from the [Compose repository release page on GitHub](#). Follow the instructions from the link, which involve running the curl command in your terminal to download the binaries. These step-by-step instructions are also included below.

2.1 Install Docker Compose

Run this command to download the current stable release of Docker Compose:

```
$ sudo curl -L "https://github.com/docker/compose/releases/download/1.24.1/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
```

2.2 Apply permissions

Apply executable permissions to the binary

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

Note: If the command docker-compose fails after installation, check your path. You can also create a symbolic link to /usr/bin or any other directory in your path.

For example:

```
$ sudo ln -s /usr/local/bin/docker-compose /usr/bin/docker-compose
```

2.3 Verify

Verify that Docker Compose is installed correctly.

```
$ docker-compose --version
```

2.4 Uninstallation

To uninstall Docker Compose if you installed using curl:

```
$ sudo rm /usr/local/bin/docker-compose
```

To uninstall Docker Compose if you installed using pip:

```
$ pip uninstall docker-compose
```

3. Chirpstack

3.1 Clone Code

Run this command to download the current stable release of Chirpstack:

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

3.2 Band and Channel Configuration

After the code is downloaded, the frequency band and channel need to be configured.

- File location: \chirpstack-docker\docker-compose-env.yml
- change the band

```
version: "3"

services:
  chirpstack-network-server:
    image: chirpstack/chirpstack-network-server:3
    environment:
      - POSTGRES__DSN=postgres://chirpstack_ns:chirpstack_ns@postgresql/chirpstack_ns?sslmode=disable
      - REDIS__URL=redis://redis:6379
      - NETWORK_SERVER__BAND_NAME=CN470
      - NETWORK_SERVER__GATEWAY__BACKEND__MQTT__SERVER=tcp://mosquitto:1883
      - JOIN_SERVER__DEFAULT__SERVER=http://chirpstack-application-server:8003
      - GEOLOCATION_SERVER__SERVER=chirpstack-geolocation-server:8005

  chirpstack-application-server:
    image: chirpstack/chirpstack-application-server:3
    ports:
      - 8080:8080
    environment:
      - POSTGRES__DSN=postgres://chirpstack_as:chirpstack_as@postgresql/chirpstack_as?sslmode=disable
      - REDIS__URL=redis://redis:6379
      - APPLICATION_SERVER__INTEGRATION__MQTT__SERVER=tcp://mosquitto:1883
      - APPLICATION_SERVER__API__PUBLIC_HOST=chirpstack-application-server:8001
      - APPLICATION_SERVER__EXTERNAL_API__JWT_SECRET=verysecret

  chirpstack-gateway-bridge:
    image: chirpstack/chirpstack-gateway-bridge:3
    ports:
      - 1700:1700/udp
    environment:
      - INTEGRATION__MQTT__AUTH__GENERIC__SERVERS=tcp://mosquitto:1883

  chirpstack-geolocation-server:
    image: chirpstack/chirpstack-geolocation-server:3
    environment:
      - GEO_SERVER__BACKEND__NAME=collos

  postgresql:
    image: postgres:9.6-alpine
    environment:
      - POSTGRES_PASSWORD=root
    volumes:
      - ./configuration/postgresql/initdb:/docker-entrypoint-initdb.d
      - postgresqldata:/var/lib/postgresql/data
```

- File location:\chirpstack-docker\configuration\chirpstack-network-server\chirpstack-network-server.toml
- Refer to the example to modify each frequency band

```
# See https://www.chirpstack.io/network-server/install/config/ for a full
# configuration example and documentation.
#
# This file is for the US915 band (channels 0-7). See the examples/ folder for more
# configuration examples.

[postgresql]
dsn="postgres://chirpstack_ns:chirpstack_ns@postgresql/chirpstack_ns?sslmode=disable"

[redis]
url="redis://redis:6379"

[network_server]
net_id="000000"

[network_server.band]
name="CN470"

[network_server.network_settings]
enabled_uplink_channels=[0, 1, 2, 3, 4, 5, 6, 7]

[network_server.gateway.backend.mqtt]
server="tcp://mosquitto:1883"

[join_server.default]
server="http://chirpstack-application-server:8003"

[geolocation_server]
server="chirpstack-geolocation-server:8005"
```

Save and run the Chirpstack by the following instructions:

```
$ docker-compose up
```

Or use this command to run ChirpStack in the background:

```
sudo docker-compose up -d
```

3.3 Port Configuration

When running software fails and the Chirpstack-gateway-bridge ports and application-server ports are occupied, the ports need to be modified.

- File location: \chirpstack-docker\docker-compose.yml
- change the chirpstack-gateway-bridge ports and application-server ports

```
version: "3"
services:
  chirpstack-network-server:
    image: chirpstack/chirpstack-network-server:3
    volumes:
      - ./configuration/chirpstack-network-server:/etc/chirpstack-network-server
  chirpstack-application-server:
    image: chirpstack/chirpstack-application-server:3
    ports:
      - 8080:8080
    volumes:
      - ./configuration/chirpstack-application-server:/etc/chirpstack-application-server
  chirpstack-gateway-bridge:
    image: chirpstack/chirpstack-gateway-bridge:3
    ports:
      - 1700:1700/udp
    volumes:
      - ./configuration/chirpstack-gateway-bridge:/etc/chirpstack-gateway-bridge
  chirpstack-geolocation-server:
    image: chirpstack/chirpstack-geolocation-server:3
    volumes:
      - ./configuration/chirpstack-geolocation-server:/etc/chirpstack-geolocation-server
  postgresql:
    image: postgres:9.6-alpine
    environment:
      - POSTGRES_PASSWORD=root
    volumes:
      - ./configuration/postgresql/initdb:/docker-entrypoint-initdb.d
      - postgresqldata:/var/lib/postgresql/data
  redis:
    image: redis:5-alpine
    volumes:
      - redisdata:/data
  mosquito:
    image: eclipse-mosquitto
    ports:
      - 1883:1883
volumes:
  postgresqldata:
  redisdata:
```

- File location: \chirpstack-docker\docker-compose-env.yml
- change the chirpstack-gateway-bridge ports and application-server ports

```

version: "3"

services:
  chirpstack-network-server:
    image: chirpstack/chirpstack-network-server:3
    environment:
      - POSTGRES_URL=postgres://chirpstack_ns:chirpstack_ns@postgres1/chirpstack_ns?sslmode=disable
      - REDIS_URL=redis://redis:6379
      - NETWORK_SERVER_BAND_NAME=CN470
      - NETWORK_SERVER_GATEWAY_BACKEND_MQTT_SERVER=tcp://mosquitto:1883
      - JOIN_SERVER_DEFAULT_SERVER=http://chirpstack-application-server:8003
      - GEOLOCATION_SERVER_SERVER=chirpstack-geolocation-server:8005

  chirpstack-application-server:
    image: chirpstack/chirpstack-application-server:3
    ports:
      - 8080:8080
    environment:
      - POSTGRES_URL=postgres://chirpstack_as:chirpstack_as@postgres1/chirpstack_as?sslmode=disable
      - REDIS_URL=redis://redis:6379
      - APPLICATION_SERVER_INTEGRATION_MQTT_SERVER=tcp://mosquitto:1883
      - APPLICATION_SERVER_API_PUBLIC_HOST=chirpstack-application-server:8001
      - APPLICATION_SERVER_EXTERNAL_API_JWT_SECRET=verysecret

  chirpstack-gateway-bridge:
    image: chirpstack/chirpstack-gateway-bridge:3
    ports:
      - 1700:1700/udp
    environment:
      - INTEGRATION_MQTT_AUTH_GENERIC_SERVERS=tcp://mosquitto:1883

  chirpstack-geolocation-server:
    image: chirpstack/chirpstack-geolocation-server:3
    environment:
      - GEO_SERVER_BACKEND_NAME=collos

  postgresql:
    image: postgres:9.6-alpine
    environment:
      - POSTGRES_PASSWORD=root
    volumes:
      - ./configuration/postgresql/initdb:/docker-entrypoint-initdb.d
      - postgresqldata:/var/lib/postgresql/data

  redis:
    image: redis:5-alpine
    volumes:
      - redisdata:/data

```

- File location: \chirpstack-docker\cnfiguration\docker-compose-env.yml
- change the chirpstack-gateway-bridge ports and application-server ports

```

# See https://www.chirpstack.io/application-server/install/config/ for a full
# configuration example and documentation.

[postgresql]
dsn="postgres://chirpstack_as:chirpstack_as@postgres1/chirpstack_as?sslmode=disable"

[redis]
url="redis://redis:6379"

[application_server.integration.mqtt]
server="tcp://mosquitto:1883"

[application_server.api]
public_host="chirpstack-application-server:8001"

[application_server.external_api]
bind="0.0.0.0:8080"
jwt_secret="verysecret"

```