Installing ThingsBoard CE on Ubuntu Server

Prerequisites

This guide describes how to install ThingsBoard on Ubuntu 20.04 LTS / 22.04 LTS / 24.04 LTS. Hardware requirements depend on chosen database and amount of devices connected to the system. To run ThingsBoard and PostgreSQL on a single machine you will need at least 4Gb of RAM. To run ThingsBoard and Cassandra on a single machine you will need at least 8Gb of RAM.

Step 1. Install Java 17 (OpenJDK)

ThingsBoard service is running on Java 17. Follow this instructions to install OpenJDK 17:

```
sudo apt update
sudo apt install openjdk-17-jdk
```


Please don't forget to configure your operating system to use Open JDK 17 by default. You can c ersion is the defa using the following command:

```
sudo update-alternatives --config java
```

You can check the installation using the following command:

```
java -version
```

Expected command output is:

```
openjdk version "17.x.xx"
OpenJDK Runtime Environment (...)
OpenJDK 64-Bit Server VM (...)
```

Step 2. ThingsBoard service installation

Download installation package.

```
wget https://github.com/thingsboard/thingsboard/releases/download/v3.8.1/thingsboard-3.8.1.deb
```

Install ThingsBoard as a service

```
sudo dpkg -i thingsboard-3.8.1.deb
```

Step 3. Configure ThingsBoard database

ThingsBoard is able to use SQL or hybrid database approach. See corresponding architecture page for more details.

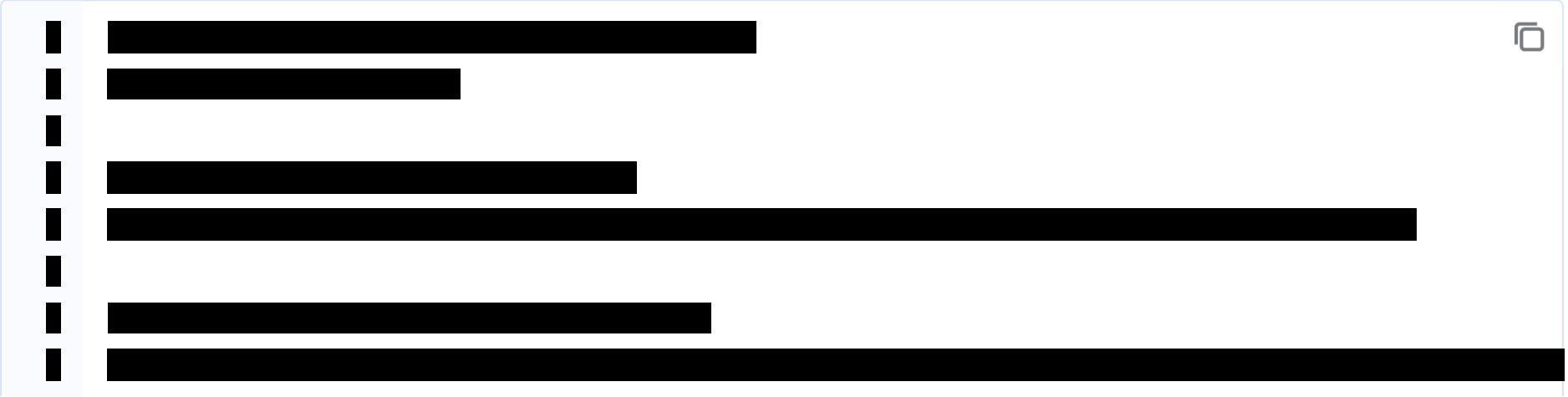
PostgreSQL
(recommended for < 5K msg/sec)

Hybrid
PostgreSQL+Cassandra
(recommended for > 5K msg/sec)

ThingsBoard team recommends to use PostgreSQL for development and production environments with reasonable load (< 5000 msg/sec). Many cloud vendors support managed PostgreSQL servers which is a cost-effective solution for most of ThingsBoard instances.

PostgreSQL Installation

Instructions listed below will help you to install PostgreSQL.



```
9
10  # install and launch the postgresql service:
11  sudo apt update
12  sudo apt -y install postgresql-15
13  sudo service postgresql start
```

Once PostgreSQL is installed you may want to create a new user or set the password for the main user. The instructions below will help to set the password for main PostgreSQL user.

To switch your current user context to the postgres user, execute the following script:

```
sudo su - postgres
```

To be able to interact with the PostgreSQL database, enter:

```
psql
```

You will connect to the database as the main PostgreSQL user. To set the password, enter the following command after **postgres=#** :

```
\password
```

Enter and confirm the password. Then, press “Ctrl+D” to return to main user console.

Then, connect to the “postgres” database as the “postgres” user:

```
psql -U postgres -d postgres -h 127.0.0.1 -W
```

Create the ThingsBoard database named “thingsboard” :

```
CREATE DATABASE thingsboard;
```

Press “Ctrl+D” twice to exit PostgreSQL.

ThingsBoard Configuration

Edit ThingsBoard configuration file

```
sudo nano /etc/thingsboard/conf/thingsboard.conf
```

Add the following lines to the configuration file. Don’t forget **to replace** “PUT_YOUR_POSTGRESQL_PASSWORD_HERE” with your **real postgres user password**:

```
# DB Configuration
export DATABASE_TS_TYPE=sql
export SPRING_DATASOURCE_URL=jdbc:postgresql://localhost:5432/thingsboard
export SPRING_DATASOURCE_USERNAME=postgres
export SPRING_DATASOURCE_PASSWORD=PUT_YOUR_POSTGRESQL_PASSWORD_HERE
# Specify partitioning size for timestamp key-value storage. Allowed values: DAYS, MONTHS, YEARS, IN
export SQL_POSTGRES_TS_KV_PARTITIONING=MONTHS
```

Step 4. Choose ThingsBoard queue service

ThingsBoard is able to use various messaging systems/brokers for storing the messages and communication between ThingsBoard services. How to choose the right queue implementation?

- **In Memory** queue implementation is built-in and default. It is useful for development(PoC) environments and is not suitable for production deployments or any sort of cluster deployments.
- **Kafka** is recommended for production deployments. This queue is used on the most of ThingsBoard production environments now. It is useful for both on-prem and private cloud deployments. It is also useful if you like to stay independent from your cloud provider. However, some providers also have managed services for Kafka. See AWS MSK for example.
- **RabbitMQ** is recommended if you don't have much load and you already have experience with this messaging system.
- **AWS SQS** is a fully managed message queuing service from AWS. Useful if you plan to deploy ThingsBoard on AWS.
- **Google Pub/Sub** is a fully managed message queuing service from Google. Useful if you plan to deploy ThingsBoard on Google Cloud.
- **Azure Service Bus** is a fully managed message queuing service from Azure. Useful if you plan to deploy ThingsBoard on Azure.
- **Confluent Cloud** is a fully managed streaming platform based on Kafka. Useful for a cloud agnostic deployments.

See corresponding architecture page and rule engine page for more details.

In Memory **(built-in and default)**	Kafka (recommended for on-prem, production installations)	Kafka in docker container (recommended for on-prem, production installations)
AWS SQS (managed service from AWS)	Google Pub/Sub (managed service from Google)	Azure Service Bus (managed service from Azure)
RabbitMQ (for small on-prem installations)	Confluent Cloud (Event Streaming Platform based on Kafka)	

In Memory queue is built-in and enabled by default. No additional configuration steps required.

Step 5. [Optional] Memory update for slow machines (4GB of RAM)

Edit ThingsBoard configuration file

```
sudo nano /etc/thingsboard/conf/thingsboard.conf
```

Add the following lines to the configuration file.

```
# Update ThingsBoard memory usage and restrict it to 2G in /etc/thingsboard/conf/thingsboard.conf
export JAVA_OPTS="$JAVA_OPTS -Xms2G -Xmx2G"
```

We recommend adjusting these parameters depending on your server resources. It should be set to at least 2G (gigabytes), and increased accordingly if there is additional RAM space available. Usually, you need to set it to 1/2 of your total RAM if you do not run any other memory-intensive processes (e.g. Cassandra), or to 1/3 otherwise.

Step 6. Run installation script

Once ThingsBoard service is installed and DB configuration is updated, you can execute the following script:

```
# --loadDemo option will load demo data: users, devices, assets, rules, widgets.
sudo /usr/share/thingsboard/bin/install/install.sh --loadDemo
```

Step 7. Start ThingsBoard service

Execute the following command to start ThingsBoard:

```
sudo service thingsboard start
```

Once started, you will be able to open Web UI using the following link:

```
http://localhost:8080/
```

The following default credentials are available if you have specified *–loadDemo* during execution of the installation

script:

- **System Administrator**: sysadmin@thingsboard.org / sysadmin
- **Tenant Administrator**: tenant@thingsboard.org / tenant
- **Customer User**: customer@thingsboard.org / customer

You can always change passwords for each account in account profile page.

Please allow up to 90 seconds for the Web UI to start.

Post-installation steps

Configure HAProxy to enable HTTPS

You may want to configure HTTPS access using HAProxy. This is possible in case you are hosting ThingsBoard in the cloud and have a valid DNS name assigned to your instance. Please follow this guide to install HAProxy and generate valid SSL certificate using Let's Encrypt.

Troubleshooting

ThingsBoard logs are stored in the following directory:

```
/var/log/thingsboard
```

You can issue the following command in order to check if there are any errors on the backend side:

```
cat /var/log/thingsboard/thingsboard.log | grep ERROR
```

Next steps

- Getting started guides - These guides provide quick overview of main ThingsBoard features. Designed to be completed in 15-30 minutes.
- Connect your device - Learn how to connect devices based on your connectivity technology or solution.
- Data visualization - These guides contain instructions on how to configure complex ThingsBoard dashboards.
- Data processing & actions - Learn how to use ThingsBoard Rule Engine.
- IoT Data analytics - Learn how to use rule engine to perform basic analytics tasks.
- Hardware samples - Learn how to connect various hardware platforms to ThingsBoard.
- Advanced features - Learn about advanced ThingsBoard features.
- Contribution and Development - Learn about contribution and development in ThingsBoard.