**THINGSBOARD SETUP (On Windows)**

1. **Prerequisite**:

Windows: version **10/8.1/8/7 32-bit/64-bit**.

Hardware requirements: Depends on database and amount of devices connected to the system. To run **ThingsBoard** and **PostgreSQL** on a single machine you will need **atleast 2GB RAM**. (In our case, we have used **16GB RAM** for handling complex functionalities like Advanced Analytics and Telemetry messages from devices)

1. **Install Java 8 (OpenJDK):**

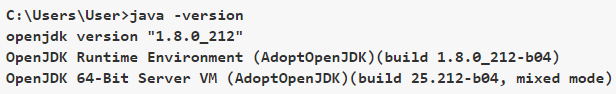
ThingsBoard service is running on Java 8. Follow the instructions to install OpenJDK 8.

* Visit [Open JDK Download Page](https://adoptopenjdk.net/index.html) to download latest **OpenJDK 8 (LTS)** MSI package.
* Run the downloaded MSI package and follow the instructions. Make sure you have selected “**Add to PATH**” and “**Set JAVA\_HOME variable**” options to “Will be installed on local hard drive” state.

You can check the installation using the following command (using Command Prompt):



Expected command output is:



1. **ThingsBoard service installation :**

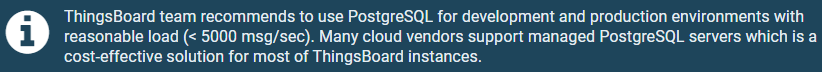
Download and extract the package.



**Note:** We assume you have extracted ThingsBoard package to default location: *C:\Program Files (x86)\thingsboard*

1. **Configure ThingsBoard database :**

ThingsBoard is able to use SQL or hybrid database approach. (Used PostgreSQL here)



PostgreSQL Installation

Download the installation file (PostgreSQL 11.7 or newer releases) [here](https://www.enterprisedb.com/downloads/postgres-postgresql-downloads#windows) and follow the installation instructions.

During PostgreSQL installation, you will be prompted for superuser (postgres) password. Don’t forget this password. It will be used later. For simplicity, we will substitute it with “postgres”.

Create ThingsBoard Database

Once installed, launch the “pgAdmin” software and login as superuser (postgres). Open your server and create database “thingsboard” with owner “postgres”.

Thingsboard Configuration

In case you have specified the PostgreSQL superuser password as “postgres”, you can skip this step. Open the Notepad or other editor as administrator user (right click on the app icon and select “Run as administrator”).

Open the following file for editing (select “All Files” instead of “Text Documents” in file choosing dialog, the encoding is UTF-8):



and locate “# SQL DAO Configuration” block. Don’t forget to replace “postgres” with your real postgres user password:



locate “SQL\_POSTGRES\_TS\_KV\_PARTITIONING” parameter in order to override the default value for timestamp key-value storage partitioning size: (not used in our case due to latest version)



1. **Choose ThingsBoard queue service :**

ThingsBoard is able to use various messaging systems/brokers for storing the messages and communication between ThingsBoard services.

**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.

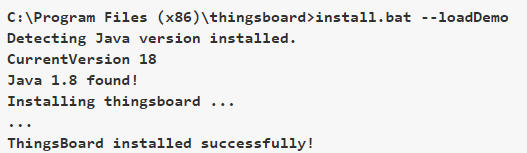
In Memory queue is built-in and enabled by default. No additional configuration steps required.

1. **Run Installation script :**

Launch windows shell (Command Prompt) as Administrator. Change directory to your ThingsBoard installation directory.

Execute **install.bat** script to install ThingsBoard as a Windows service (or run **“install.bat –loadDemo”** to install and add demo data). This means it will be automatically started on system startup. Similar, **uninstall.bat** will remove ThingsBoard from Windows services.

The output should be similar to this one:



1. **Running ThingsBoard service :**

Now let’s start the ThingsBoard service! Open the command prompt as an Administrator and execute the following command:



Expected output:



In order to restart the ThingsBoard service you can execute following commands:

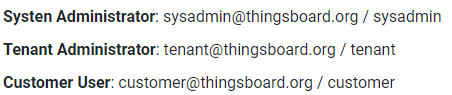


Once started, you will be able to open Web UI using the following link:

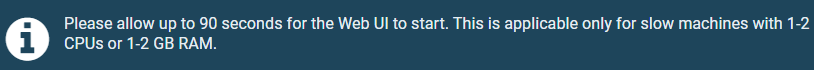


**Note:** If you are not able to see the Web UI then it means that the port is utilized by the system for some other process. So, either you can kill the process running on this port or if the process is a system kernel then, in the ThingsBoard config file, change the PORT to any other. [In our case, the Web UI link was available as ***http://localhost:8083/*** ]

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



**Note:** You can change the credentials after logging into the Web UI in the account profile section.



1. **Troubleshooting :**

The log files are located in **logs** folder (“C:\Program Files (x86)\thingsboard\logs” in our case).

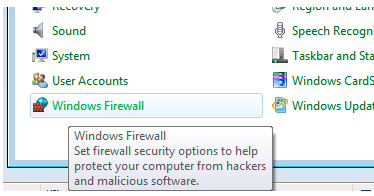
The **thingsboard.log** file should contain following line:

**YYYY-MM-DD HH:mm:ss,sss [main] INFO o.t.s.ThingsboardServerApplication - Started ThingsboardServerApplication in x.xxx seconds (JVM running for x.xxx)**

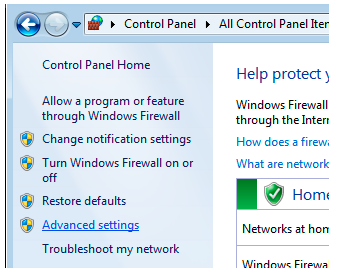
1. **Windows Firewall Settings :**

In order to have external access to ThingsBoard Web UI and device connectivity (HTTP, MQTT, CoAP) you need to create a new inbound rule with Windows Firewall with Advanced Security.

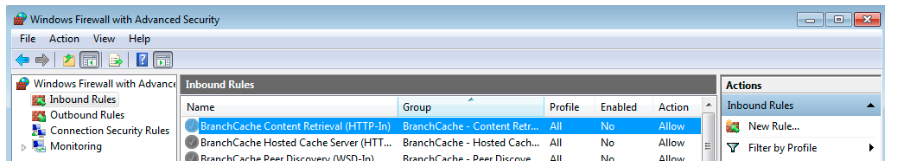
* Open “Windows Firewall” from “Control Panel”:



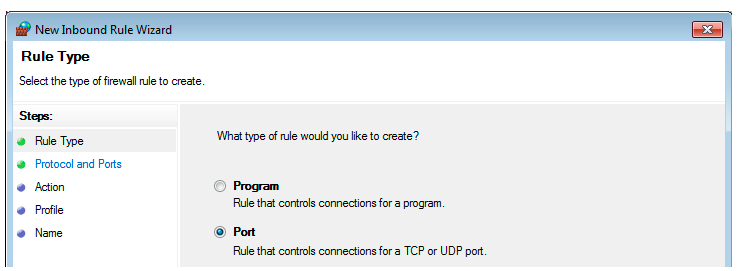
* Click “Advanced settings” on the left panel:



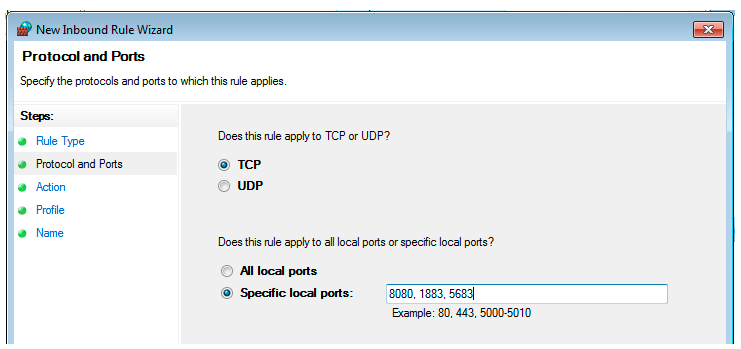
* Select “Inbound Rules” on the left panel, then click “New Rule…” on the right “Actions” panel:



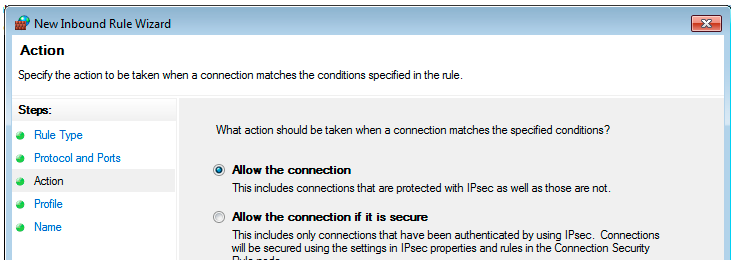
* Now new “New Inbound Rule Wizard” window will open. On the first step “Rule Type” select “Port” option:



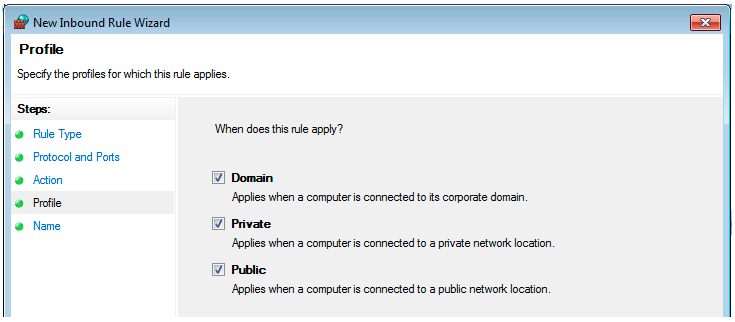
* On the “Protocol and Ports” step select “TCP” protocol and enter port list **8080 (can be your configured port for HTTP), 1883, 5683** in the “Specific local ports” field:



* On the “Action” step leave “Allow the connection” option selected:



* On the “Profile” step select Windows network profiles when to apply this rule:



* Finally, give the name to this rule (for ex. “ThingsBoard Service Networking”) and click “Finish”.

