# Manuale d’installazione e d’uso

L’installazione di tutte le componenti necessarie per il corretto funzionamento del codice richiede una lunga serie di passaggi che devono essere svolti principalmente da linea di comando.

I primi passaggi di questa installazione verranno svolti su Ubuntu, un programma scaricabile da “Microsoft store” che consiste in un sistema operativo basato su Linux, e, successivamente, ci si sposterà sulla “Windows Subsystem for Linux”.

Requisiti software

Per prima cosa bisogna scaricare “Ubuntu”.

Una volta fatto ciò, si deve avviare l’applicazione e, alla prima apertura, verrà richiesto di creare una partizione Linux tramite un “username” e una “password”.

Questa password sarà fondamentale perché verrà richiesta ogni volta che si deciderà di far ripartire l’application server.

Successivamente bisogna installare WSL 2.0 sulla propria macchina.

***https://www.omgubuntu.co.uk/how-to-install-wsl2-on-windows-10***

Prima di fare ciò, è necessario aprire il “Prompt dei comandi” in modalità “Esegui come amministratore” ed eseguire il comando

* wsl.exe –install

per abilitare le funzioni opzionali di WSL.

Fatto questo, bisognerà riavviare il sistema, aprire Ubuntu ed eseguire i seguenti due comandi

* sudo apt update
* sudo apt upgrade

A questo punto si può passare all’installazione di WSL 2.0. Per eseguire l’upgrade da WSL 1 basta aprire nuovamente il “Promp dei comandi” con i permessi d’amministratore ed eseguire il comando

* wsl.exe --set-version Ubuntu 2

In questo modo si installerà e si setterà di default WSL 2.0 sul proprio computer.

Requisiti per le componenti del ChirpStack LoRaWAN Network Server

***https://www.chirpstack.io/project/install/requirements/***

Per prima cosa bisogna aprire Ubuntu ed eseguire i seguenti 3 comandi

* sudo apt install mosquitto

Questo comando installa Mosquitto, un broker dei messaggi che implementa il protocollo MQTT

* sudo apt install postgresql

Questo comando installa PostgreSQL, un DBMS (DataBase Management System)

* sudo apt install redis-server

Questo commando installa Redis, un archivio di strutture dati in memoria open source usato come database, cache e broker di messaggi.

Installazione application server

***https://www.chirpstack.io/application-server/install/debian/***

A questo punto si può passare all’installazione dei componenti messi a disposizione dallo stack ChirpStack.

I seguenti comandi verranno eseguiti su Ubuntu con i permessi d’amministratore:

* sudo -u postgres psql

Per creare un database per l’application server e spostarsi nel prompt di PostgreSQL

* create role chirpstack\_as with login password ‘dbpassword’

Per creare l’user chirpstack\_as (inserire una password al posto di ‘dbpassword’).

* create database chirpstack\_as with owner chirpstack\_as

Per creare il database chirpstack\_as

* \c chirpstack\_as
* create extension pg\_trgm
* create extension hstore
* \q

Per verificare il corretto setup, si può eseguire il comando

* psql -h localhost -U chirpstack\_as -W chirpstack\_as

I successivi comandi sono usati per installare ed avviare le varie componenti:

* sudo apt install chirpstack-gateway-bridge

sudo systemctl start chirpstack-gateway-bridge

* sudo apt install chirpstack-network-server

sudo systemctl start chirpstack-network-server

* sudo apt install chirpstack-application-server

sudo systemctl start chirpstack-application-server

“start” serve per avviare la particolare componente. Se al suo posto si scrive “restart” o “stop” o “status”, il comando, rispettivamente, esegue un riavvio o termina l’esecuzione o fornisce lo stato del componente.

I seguenti comandi non servono per l’installazione delle componenti, ma permettono di visualizzare a video l’output di quella particolare componente:

* journalctl -u chirpstack-gateway-bridge -f -n 50
* journalctl -u chirpstack-network-server -f -n 50
* journalctl -u chirpstack-application-server -f -n 50

Quando si crea un componente, viene generato il suo corrispondente “Configuration file”. Per potervi accedere bisogna eseguire i comandi che seguono:

* sudo su

Per ottenere i permessi di amministratore su Ubuntu

* cd /etc

Per accedere alla directory contenente le cartelle relative ai vari componenti.

Dentro questa cartella bisogna eseguire il comando

* cd /’componente’

dove al posto di ‘componente’ bisogna scrivere o chirpstack-gateway-bridge o chirpstack-network-server o chirpstack-application-server per entrare nella cartella di quello specifico componente. Infine, per aprire il file contenente la configurazione di quel particolare componente bisogna eseguire il comando

* sudo nano ‘componente’.toml

A questo punto si devono eseguire i seguenti tre comandi:

* cd /tmp
* wget --content-disposition \https://gist.githubusercontent.com/djfdyuruiry/6720faa3f9fc59bfdf6284ee1f41f950/raw/952347f805045ba0e6ef7868b18f4a9a8dd2e47a/install-sg.sh chmod +x /tmp/install-sg.sh
* /tmp/install-sg.sh && rm /tmp/install-sg.sh

Fatto ciò si deve aprire il “Prompt dei comandi” tramite “Esegui come amministratore” ed eseguire questi altri tre comandi:

* wsl –shutdown

Per terminare l’esecuzione di Ubuntu

* wsl genie -s
* sudo systemctl status time-sync.target

In questo modo l’interfaccia grafica dell’application server sarà visibile all’indirizzo

* localhost:8080

Una volta spento il computer, per poter visualizzare nuovamente l’application server in locale sarà necessario eseguire il comando

* wsl genie -s

dal “Prompt dei comandi” (avviato sempre con “Esegui come amministratore”), altrimenti cercando direttamente “localhost:8080” il motore di ricerca dirà che a tale indirizzo non è possibile trovare nulla.

Piccola nota su questo comando: se dovessero vedersi a video una serie di “!” in numero sempre crescente, vuol dire che il comando non funzionerà.

Il trucco sta nell’interrompere l’esecuzione del comando (tramite la combinazione di comandi “CTRL” + ”c”) ed eseguirlo nuovamente.

Configurazione dell’interfaccia grafica

Una volta inseriti “username” e “password” ed entrati nell’application server, bisogna passare alla configurazione della rete.

Organization

Per creare una nuova “Organization” (se non già presente o se si vuole configurare tutto da capo), cliccare su “Organizations” sulla sinistra dell’interfaccia ed inserire “Name” e “Display name”, mantenendo a 0 “Number of devices” e spuntando la voce “Organization can have gateways”.

Applications

Network-servers

Gateway-profiles

Gateway

Device-profile

Device in applications

Modifiche da apportare al codice sorgente

Configurato il tutto, se si prova a far girare il codice sorgente esso non funzionerà. Infatti bisogna apportare delle modifiche al codice ogni qual volta che si esegue il comando per avviare l’application server.

Innanzitutto bisogna accedere alla macchina virtuale che gira si WSL 2.0 dal codice: questo creerà la connessione tra le componenti del codice e quelle create da linea di comando.

***https://www.youtube.com/watch?v=yCK3easuYm4&ab\_channel=DavidBombal***

Su Ubuntu eseguire i seguenti comandi

* wsl -l -v

Per vedere la verisone id Ubuntu running

* ip addr | grep eth0

Per prelevare l’indirizzo ip della virtual machine (indirizzo ip dopo la voce “inet”)

Per vedere l’indirizzo ip di Windows, eseguire su “Prompt dei comandi”

* ipconfig

e guardare sotto la sezione “Scheda LAN wireless Wi-Fi indirizzo IPv4”.

Arrivati a questo punto bisogna consentire la connessione da Ubuntu a Windows. Questo viene fatto attraverso due passaggi.

Il primo consiste nel creare una nuova regola sul firewall per gestire la connessione di dispositivi diversi alla macchina virtuale Ubuntu. Infatti, a questo punto, l’unico dispositivo che conosce della Virtual Machine che gira su Ubuntu è il computer sul quale Ubuntu è scaricato. Bisogna rendere quella Virtual Machine visibile anche ad altri dispositivi. Per fare ciò bisogna seguire questi passaggi:

Pannello di controllo --> Rete internet --> Centro connessioni di rete --> Windows defender firewall -> impostazioni avanzate --> Regole connessioni in entrata --> Nuova regola (con tasto dx) --> Selezionare porta, avanti --> mettere porta 3390(o una qualsiasi libera) --> avanti fino alla fine

Il secondo passo, invece, deve essere eseguito da “Windows PowerShell” dotato dei permessi d’amministratore. Bisogna eseguire il comando

* netsh interface portproxy add v4tov4 listenport=3390 listenaddress=0.0.0.0 connectport=’port’ connectaddress=’ubuntu\_addr’

Dove bisogna sostituire ‘port’ con il numero di porta che si è inserito al passo precedente (es.: 3390) e ‘ubuntu\_addr’ con l’indirizzo della Virtual Machine.

Questi passi andranno eseguiti tutti una sola volta, tranne quello che consente di prelevare l’indirizzo della Virtual Machine da Ubuntu: tale indirizzo, infatti, andrà poi inserito nel codice in un determinato punto.

Successivamente, bisognerà prelevare dall’interfaccia web tre chiavi

* Application Session Key (OTAA)
* Application Session Key (Activation)
* Network Session Key (Activation)

Ed inserirle nel codice sorgente.

Se tutti questi passaggi sono stati fatti correttamente, il codice comunica con l’interfaccia grafica dell’application server e sarà possibile vedere i messaggi.