

DESIRA chatbot - Manuale utente

2021

Contents

1	Introduzione	3
2	Installazione	4
2.1	Prerequisiti	4
2.2	Botpress	5
2.3	Intermediario	5
2.4	MongoDB	6
3	Configurazione	6
3.1	PM2	6
3.2	Botpress	6
3.3	Intermediario	7
3.4	MongoDB	7
3.5	Ultimi passi	8

1 Introduzione

L'obiettivo di questo progetto è fornire una semplice interfaccia conversazionale per la creazione di segnalazioni relative a richieste di intervento nel contesto della mitigazione del dissesto idrogeologico. Il sistema è costituito da quattro componenti principali:

1. Un'istanza di Botpress per la realizzazione della logica di conversazione;
2. Un programma in Python che agevola la comunicazione tra l'istanza di Botpress e Telegram agendo da intermediario;
3. Un'istanza di MongoDB per la persistenza dei dati;
4. Un'interfaccia web per il monitoraggio delle segnalazioni da parte di operatori autorizzati.

Questo manuale fornisce istruzioni per l'installazione e configurazione dei primi tre.

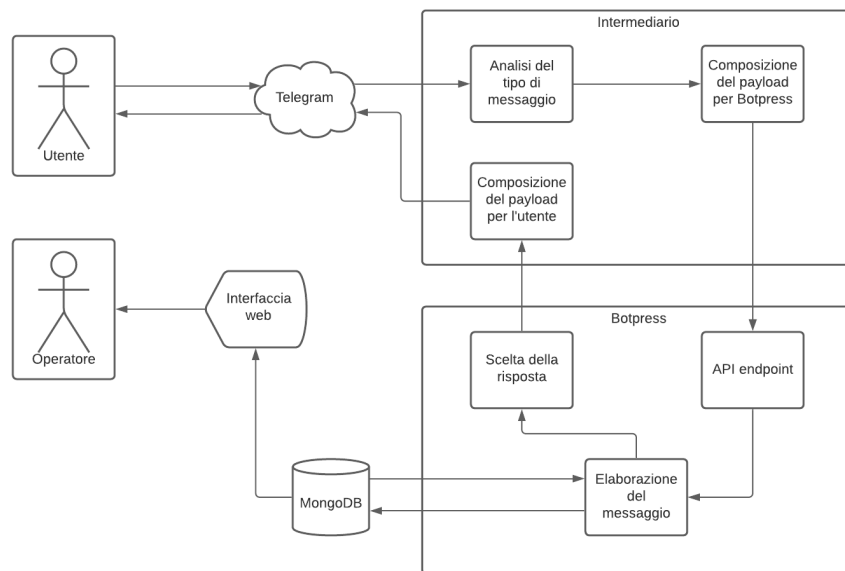


Fig. 1: Struttura generale del sistema

2 Installazione

In questa sezione vedremo come installare i vari componenti del sistema.

2.1 Prerequisiti

Per l'installazione sono necessari i seguenti pacchetti, ottenibili tramite il package manager del proprio sistema operativo:

- `git`
- `curl`
- `zip`
- `python3`
- `python3-venv`
- `python3-pip`

Ad esempio, per aggiornare l'elenco dei pacchetti e installare `git` su Ubuntu 18.04:

```
sudo apt update
sudo apt install git
```

È inoltre necessario installare Node Version Manager (`nvm`), che può essere ottenuto con il seguente comando, spezzato su più righe esclusivamente per motivi di spazio:

```
curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.39.0/
install.sh | bash
```

Potrebbe essere necessario riavviare il terminale prima di poter iniziare a utilizzare `nvm`.

Per la fase di build di Botpress necessitiamo della versione 12.18.1 dell'ambiente di esecuzione Node.js, ottenibile utilizzando il comando `nvm` appena installato:

```
nvm install 12.18.1
```

Se la suddetta versione di Node dovesse già essere presente sul sistema, il comando la imposterà automaticamente come versione attiva.

Ci servirà anche `yarn`, un gestore di pacchetti per Node.js installabile con `npm`:

```
npm install -g yarn
```

2.2 Botpress

Per lo sviluppo di questo progetto è stata utilizzata la versione 12.25.0 di Botpress, il cui codice sorgente è ottenibile da GitHub utilizzando il comando `git`:

```
git clone https://github.com/botpress/botpress.git
cd botpress
git checkout 56eaa351caac89935fce0c407ca672db83e44ce0
```

Prima di procedere alla fase di build dobbiamo aggiungere due pacchetti di Node necessari per il funzionamento del bot. Navighiamo fino al file

```
botpress/packages/bp/package.json
```

scendiamo alla categoria `dependencies` e aggiungiamo i seguenti campi:

```
"mongodb": "^4.1.0",
"libphonenumber-js": "^1.9.34"
```

Fatto ciò, torniamo nella cartella `botpress` ed eseguiamo i seguenti comandi, ignorando eventuali errori sollevati dal primo:

```
yarn cache clean
yarn
yarn build
```

Infine, creiamo un file eseguibile:

```
yarn package --linux
```

Il file verrà creato in `botpress/packages/bp/binaries/` con il nome `bp`.

2.3 Intermediario

Il codice per l'intermediario e la lista di dipendenze sono ottenibili da GitHub utilizzando il comando `git`:

```
git clone https://github.com/Nargaruga/Botpress-Middleman.git
```

Per evitare conflitti di versione dei pacchetti, prima di installare le dipendenze creiamo un ambiente virtuale utilizzando il comando `venv`. Avendo cura di trovarci nella cartella creata dal comando `git clone` appena eseguito, eseguiamo i seguenti comandi, sostituendo a "my-env" il nome che intendiamo dare all'ambiente:

```
python3 -m venv my-env
source my-env/bin/activate
```

Ora ci troviamo nell'ambiente virtuale e possiamo procedere all'installazione delle dipendenze:

```
pip install -r requirements.txt
```

Dopodichè usciamo dall'ambiente virtuale con

```
deactivate
```

2.4 MongoDB

Fare riferimento alla documentazione ufficiale di MongoDB per istruzioni su come installarlo sul proprio sistema operativo.

3 Configurazione

In questa sezione vedremo come configurare i vari componenti per garantirne il corretto funzionamento. Vedremo inoltre come fare in modo che rimangano sempre in esecuzione, anche in caso di riavvio della macchina.

3.1 PM2

PM2 è un gestore di processi che ci permette di controllare l'esecuzione delle nostre applicazioni e automatizzarne il riavvio in caso di reboot o crash. Può essere installato con `npm`:

```
npm install -g pm2
```

Eseguiamo poi il seguente comando per fare in modo che PM2 si avvii automaticamente all'accensione:

```
pm2 startup
```

3.2 Botpress

Otteniamo il codice del bot da GitHub e archiviamolo nel seguente modo:

```
git clone https://github.com/Nargaruga/DESIRA-Chatbot.git
zip -r DESIRA-Chatbot.zip DESIRA-Chatbot
rm -rf DESIRA-Chatbot
```

Avviamo Botpress con il comando `pm2`, sostituendo a "my-botpress" il nome che vogliamo associare al processo:

```
cd botpress/packages/bp/binaries
pm2 start "./bp start -p" --name my-botpress
```

Il flag `-p` avvia Botpress in production mode, con logging ridotto. A questo punto possiamo collegarci a `http://localhost:3000`, ovvero l'URL default per la dashboard di Botpress, e seguire le istruzioni a schermo per importare il bot dall'archivio `DESIRA-Chatbot.zip` creato in precedenza. Affinchè il bot funzioni, è necessario configurare tre parametri:

- URI del database;
- Nome del database;
- Path di una cartella utilizzabile dal bot per il salvataggio temporaneo delle immagini.

Per farlo apriamo la dashboard, clicchiamo sul nome del bot e poi sull'icona del code editor, situata nel menù verticale a sinistra.



Fig. 2: Icona del code editor

Da qui espandiamo la sezione **Hooks** e apriamo il seguente file:

```
after_bot_mount/initial_config.js
```

Questo file ospita tre variabili:

- `DB_URI`
- `DB_NAME`
- `IMAGE_PATH`

Per i valori da assegnare alle prime due, fare riferimento alla sezione 3.4.

Dopo aver configurato queste variabili, salviamo il file e riavviamo Botpress con questo comando, sostituendo a "my-botpress" il nome che abbiamo passato all'opzione `--name` di `pm2` all'inizio della sezione:

```
pm2 reload my-botpress
```

3.3 Intermediario

L'intermediario comunica direttamente con Telegram, pertanto dobbiamo permettergli di accedere al token rilasciato da Telegram stesso per il bot. Per farlo è sufficiente mettere nella stessa cartella di `botpress_middleman.py` un file `token.txt` contenente il token.

Per aggiungere l'intermediario alla lista di processi gestiti da PM2, rechiamoci nella cartella contenente i suoi file ed eseguiamo il seguente comando, sostituendo a "my-name" il nome che vogliamo assegnare al processo e a "my-env" il nome che abbiamo dato, in fase di installazione, alla cartella creata con `venv`:

```
pm2 start botpress_middleman.py --name my-name  
--interpreter=my-env/bin/python
```

Nota bene: si tratta di un comando singolo, qui appare su due righe esclusivamente per motivi di spazio.

3.4 MongoDB

Eseguendo il seguente comando possiamo scoprire l'init system utilizzato dal nostro sistema operativo, che ci servirà per fare in modo che MongoDB si avvii automaticamente all'accensione:

```
ps --no-headers -o comm 1
```

Se l'output del suddetto comando è `systemd`, eseguiamo i seguenti comandi:

```
sudo systemctl daemon-reload
sudo systemctl start mongod
sudo systemctl enable mongod
```

Invece, se l'output è `init`, è sufficiente eseguire il seguente comando:

```
sudo service mongod start
```

Per funzionare, il bot ha bisogno di un database Mongo con almeno due collection, chiamate "users" e "reports". Per creare un nuovo database con le due collezioni sopracitate, possiamo utilizzare i seguenti comandi, sostituendo a "my-db" il nome che si intende dare al database:

```
mongosh
use my-db
db.createCollection("users")
db.createCollection("reports")
```

Per uscire dalla console aperta con `mongosh` è sufficiente premere **CTRL+D**.

Ora abbiamo i valori da assegnare alle variabili `DB_NAME` e `DB_URI`, menzionate nella sezione 3.2: a `DB_NAME` assegnamo il nome utilizzato in `use my-db`, mentre a `DB_URI` assegnamo `mongodb://localhost/`, sempre ricordandosi di sostituire a "my-db" l'effettivo nome del database.

Nota: l'URI proposta in questa guida è la forma più basilare e serve a connettersi ad un'istanza locale del DB esposta sulla porta default e senza autenticazione. Nel caso questa configurazione non dovesse soddisfare le vostre esigenze, consigliamo la consultazione della documentazione ufficiale di MongoDB.

3.5 Ultimi passi

Verifichiamo che Botpress e l'intermediario siano stati correttamente avviati sotto PM2:

```
pm2 list
```

Se tutto è in regola, utilizziamo il seguente comando per salvare la configurazione e fare in modo che venga ripristinata all'avvio:

```
pm2 save
```

Questo conclude la configurazione.