

# Guida allo sviluppo locale

Matteo Casonato

Matteo Midenà

19 luglio 2022

# Indice

<b>1</b>	<b>Introduzione</b>	<b>3</b>
<b>2</b>	<b>Besu Private Network</b>	<b>3</b>
2.1	Prerequisiti . . . . .	3
2.1.1	Configurazione macchina . . . . .	3
2.1.2	Configurazione hardware Virtual Machine . . . . .	3
2.2	Avvio della blockchain in locale . . . . .	4
2.2.1	Setup e avvio della blockchain . . . . .	4
2.2.2	Come accedere ai servizi della blockchain . . . . .	5
2.3	Interazione con la blockchain . . . . .	5
2.3.1	Preparazione Metamask e scambio fondi . . . . .	5

# 1 Introduzione

Lo scopo del documento è quello di documentare i passi effettuati durante lo sviluppo di una soluzione locale al problema riportato nel documento "Studio del Problema e Fattibilità".

## 2 Besu Private Network

Come descritto nel documento "Analisi del Problema e Fattibilità" è stato scelto di eseguire in locale una rete privata, adibita al testing degli smart contracts. Nello specifico, si è fatto riferimento alla [guida ufficiale](#) di Hyperledger Besu, e ne riportiamo le parti più importanti.

### 2.1 Prerequisiti

Di seguito sono elencati i software che dovranno essere installati nella Virtual Machine per la corretta esecuzione dei processi di rete, interazione con essa e sviluppo di smart contracts.

#### 2.1.1 Configurazione macchina

- Uno dei seguenti sistemi operativi:
  - Linux con architettura x86\_64 (scelto Debian 11 per la nostra macchina)
  - macOS con processore Intel
  - Windows 64-bit con:
    - \* WSL2
    - \* Docker desktop configurato per funzionare su WSL2
- Docker e Docker Compose
- Node.js (versione 12 o superiore)
- Truffle
- cURL command line
- Metamask

#### 2.1.2 Configurazione hardware Virtual Machine

La configurazione dettagliata è riportata sulla documentazione di [Besu](#). Riportiamo qui solo quella utilizzata per la nostra configurazione:

- Memoria: 8GB (allocazione minima 4GB)
- Processori: 2 sockets, 12 cores
- Hard disk: 100GB (interfaccia SCSI)

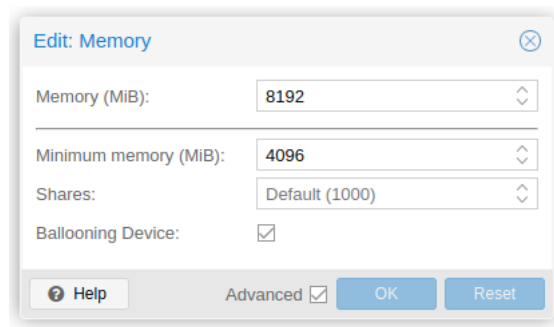


Figura 1: Allocazione memoria su Proxmox

## 2.2 Avvio della blockchain in locale

Di seguito è riportata la procedura per avviare la blockchain locale e le modalità di interazione con i vari servizi.

### 2.2.1 Setup e avvio della blockchain

Come prima cosa occorrerà scaricare tutto il necessario, e basterà eseguire il comando

```
npx quorum-dev-quickstart
```

Inizierà la procedura guidata per scegliere il tipo di rete da impostare. Selezionare le stesse risposte scelte nella seguente immagine:

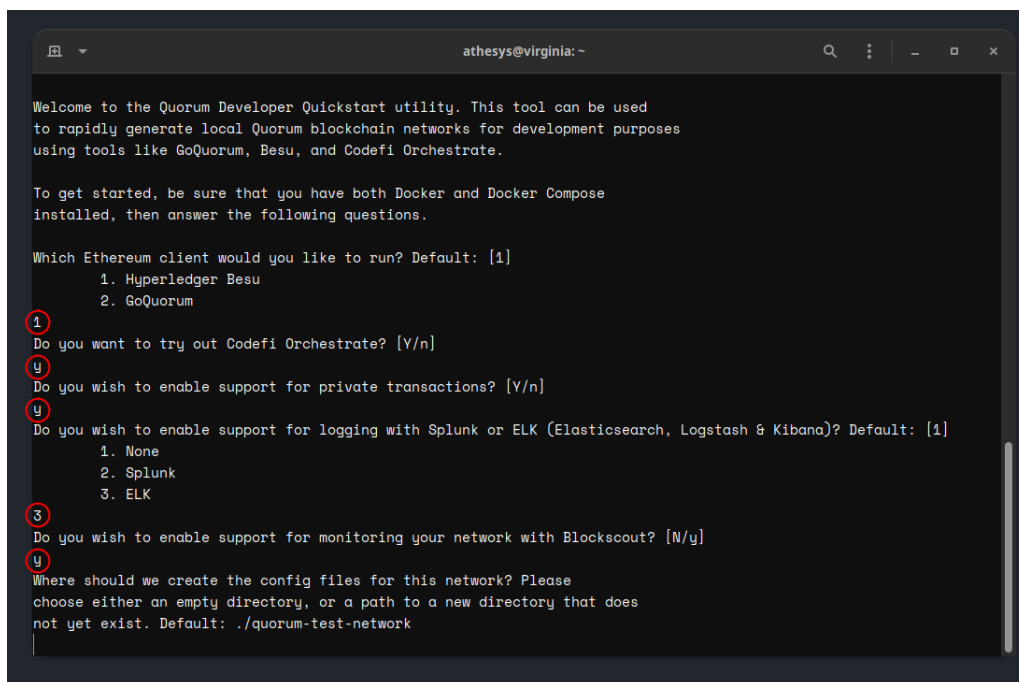


Figura 2: Setup della rete locale privata

In questo modo stiamo configurando una rete Hyperledger Besu, con Codefi Orchestrate come manager delle transazioni (più efficiente), transazioni private per emulare ancora meglio l'ambiente di produzione, ELK come logger di eventi e possibilità di utilizzo di Blockscout come block explorer (di default le transazioni sono gratuite su una rete permissioned, ma il parametro è modificabile).

Una volta generati i file necessari, occorrerà spostarsi dentro la directory appena creata, eseguire il comando `npm install` e `npm start` per eseguire gli script necessari all'avvio della rete.

### 2.2.2 Come accedere ai servizi della blockchain

Quando la rete sarà online si potrà accedere ai servizi che offre:

- **JSON-RPC HTTP** service endpoint: <http://localhost:8545>
- **JSON-RPC WebSocket** service endpoint: <ws://localhost:8546>
- **Web block explorer** address: <http://localhost:25000>
- **Prometheus** address: <http://localhost:9090/graph>
- **Grafana** address: <http://localhost:3000/d/XE4V0WGZz/besu-overview?orgId=1&refresh=10s&from=now-30m&to=now&var-system=All>
- **Kibana** logs address: <http://localhost:5601/app/kibana#/discover>

Per poterci accedere direttamente dal browser è necessario aprire il tunneling ssh, tramite il comando:

```
ssh -L chosenPort:localhost:chosenPort  
athesys@handler.bunnies.athesys.it -p22104 -i /path/to/ssh/key
```

dove il parametro `chosenPort` va sostituito con la porta scelta (ad esempio **25000** se ci si vuole connettere al block explorer), e `/path/to/ssh/key` va sostituito col percorso dove si trova la propria chiave privata usata per la connessione ssh.

## 2.3 Interazione con la blockchain

Descriviamo ora come collegare Metamask alla blockchain locale ed interagire con essa.

### 2.3.1 Preparazione Metamask e scambio fondi

Per poter utilizzare Metamask e interagire con la rete occorre creare il tunneling per il servizio **JSON-RPC HTTP** e per il **block explorer**, rispettivamente per le port **8545** e **25000**.

Ora, su Metamask occorrerà collegarsi alla **rete** corretta, ossia Localhost:8545, come nell'immagine:

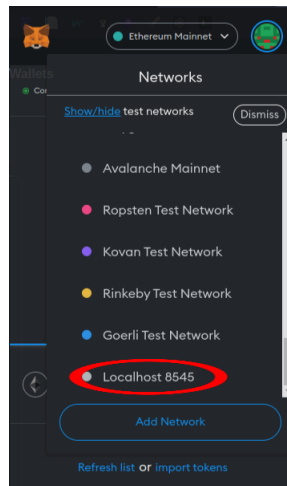


Figura 3: Scelta della rete corretta su Metamask

Inoltre occorrerà importare su Metamask uno degli **account di test** fornitoci dalla [guida ufficiale](#) (private key: 0x8f2a55949038a9610f50fb23b5883af3b4ecb3c3bb792cbcefb1542c692be63), che sarà già carico di ETH.

Una volta impostato correttamente Metamask, si potrà visitare la pagina <http://localhost:25000/explorer/wallets> e **inviare degli ETH di test** ad un altro account in possesso, per testare la rete ed eventualmente andare a visualizzare la transazione nell'explorer.