

Configuration Manual

MSc Research Project
MSC In Cybersecurity

Abhinandan Mahanta
Student ID: 23315024

School of Computing
National College of Ireland

Supervisor: Niall Heffernan

National College of Ireland
MSc Project Submission Sheet
School of Computing



Student Name: Abhinandan Mahanta
Student ID: x23315024
Programme: MSC In Cybersecurity **Year:** 2024-25
Module: MSc (Research) Practicum/Internship Part 2
Lecturer: Niall Heffernan
Submission Due Date: 15-09-2025
Project Title: Blockchain based IOT Protection
Word Count: 357 **Page Count:** 06

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Abhinandan Mahanta

Date: 15-09-2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Abhinandan Mahanta
Student ID: x23315024

1 System Requirements

1.1 Hardware

- CPU: Intel® Core™ i7 or equivalent (4 cores, 8 threads recommended)
- RAM: 8 GB minimum (16 GB recommended)
- Storage: 20 GB free space
- Network: LAN or localhost testing environment

1.2 Software

- Host OS: Windows / Linux / macOS (for Oracle VM)
- Virtual Machine: Oracle VirtualBox (≥ 7.0)
- Guest OS: Ubuntu 22.04 LTS (64-bit)
- Node.js: v18.17.0
- npm: v9.x or higher
- Git: v2.34 or higher
- htop, pidstat (sysstat package) for performance monitoring

2 Virtual Machine Setup

1. **Download & Install Oracle VirtualBox** from:
<https://www.virtualbox.org/>
2. **Download Ubuntu 22.04 LTS ISO:**
<https://ubuntu.com/download/desktop>
3. **Create a New VM** in VirtualBox:
 - Name: BlockchainIoT
 - Type: Linux, Version: Ubuntu (64-bit)
 - Memory: 8 GB
 - Storage: 50 GB dynamically allocated VDI
4. **Attach ISO** to VM and install Ubuntu.

3 Installing Dependencies

Run the following in Ubuntu terminal:

```

sudo apt update && sudo apt upgrade -y
sudo apt install git curl htop sysstat -y
curl -fsSL https://deb.nodesource.com/setup_18.x | sudo -E bash -
sudo apt install -y nodejs
node -v
npm -v

```

4 Project Setup

4.1 Open the project code folder

```

alpha@alpha-VirtualBoxx:~/Blockchain-of-Things$ tree
.
├── blockchain
│   ├── docker-compose.yml
│   ├── Dockerfile
│   ├── License.txt
│   ├── main.js
│   ├── naivechain_blockchain.png
│   ├── naivechain_components.png
│   ├── package.json
│   ├── package-lock.json
│   └── README.md
├── dashboard
│   └── sensor1.html
├── fog_server
│   ├── db
│   │   └── mongoose.js
│   ├── index.js
│   ├── models
│   │   ├── limits.js
│   │   ├── sensor1.js
│   │   └── sensor.js
│   ├── mongos.conf
│   ├── package.json
│   ├── package-lock.json
│   └── utility
│       ├── addSensor1.js
│       ├── addSensor.js
│       ├── addToBlockchain.js
│       └── setLimits.js
├── package-lock.json
├── README.md
├── ThreatDragonModels
│   └── Package Distribution
│       └── Package Distribution.json
└── 8 directories, 25 files

```

Figure 1: Project folder structure

4.2 Install Node.js Modules

```
npm install
```

5 Running the Blockchain Node

Node start

```
alpha@alpha-VirtualBoxx:~/Blockchain-of-Things/blockchain$ npm start

> naivechain@1.0.0 start
> node main.js

listening websocket p2p port on: 6001
Listening http on 127.0.0.1:3001
```

Figure 2: Running Blockchain node

The blockchain REST API will run at:

- **View Blockchain:** GET <http://127.0.0.1:3001/blocks>

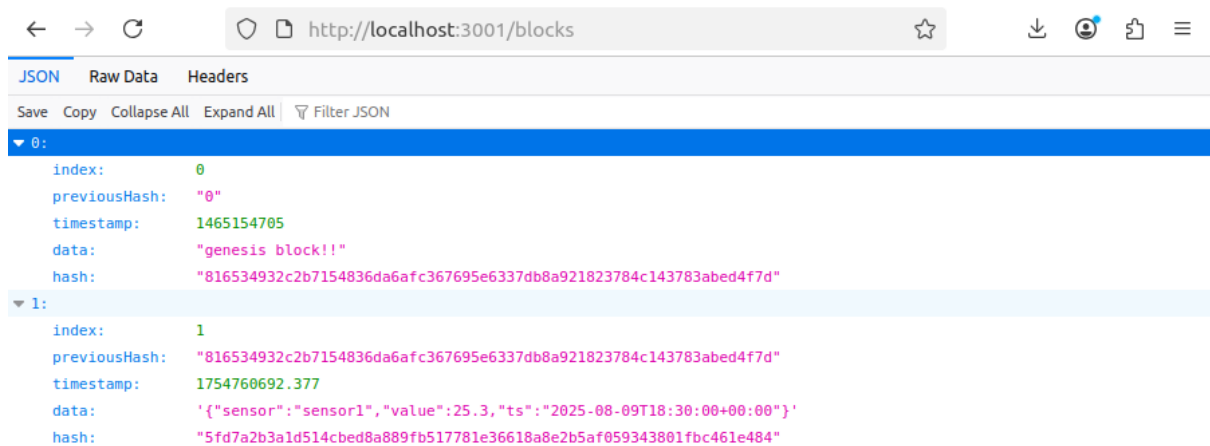


Figure 3: Blocks

Example command to send IoT data:

```
curl -s -H "Content-Type: application/json" \
-d '{"data": {"sensor": "sensor1", "value": 25.3, "ts": "2025-08-09T18:30:00+00:00"}'}
```

6 Testing & Performance Monitoring

- **Throughput Test Script, on blockchain directory**

```
./test.sh
```

- **Monitor Resource Usage**

```
htop
pidstat -u -r -p <PID>
```

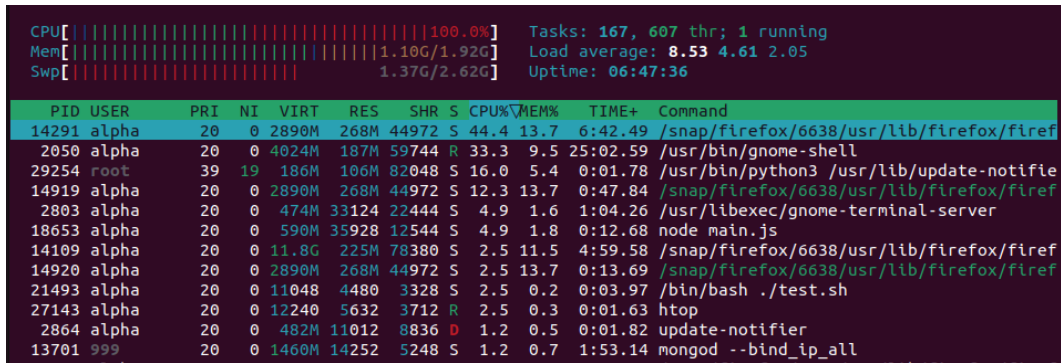


Figure 4: htop sample results

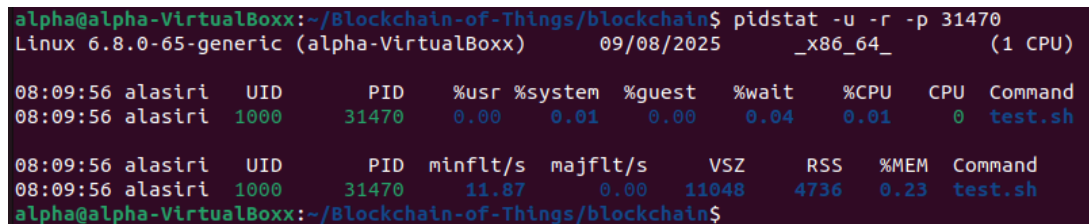


Figure 5: PIDSTAT results

7 Security Testing

- Modify block payload after mining → Validate rejection
- Replay old payload with same timestamp → Detect replay prevention
- Use invalid HMAC key → Reject unauthorized client

8 Stopping the Node

Press CTRL + C in the terminal running main.js.

9 Troubleshooting

- **Port in use** - Change port in main.js
- **Module not found** - Run `npm install` again
- **VM performance slow** - Increase RAM/CPU allocation in VirtualBox settings