

MIT WORLD PEACE UNIVERSITY

Attack Research and Documentation
Fourth Year B. Tech, Semester 8

GANACHE IN BLOCKCHAIN DEVELOPEMENT

LAB ASSIGNMENT 3

Prepared By

Krishnaraj Thadesar
Cyber Security and Forensics
Batch A1, PA 15

February 28, 2025

Contents

1 Aim	1
2 Demo	1
3 Frequently Asked Questions	1
3.1 What is Ganache and how is it used in blockchain development?	1
3.2 What are the key differences between Ganache test network and the real Ethereum network structure?	2
3.3 How can you deploy a Solidity smart contract to a Ganache network using Truffle? . . .	2
3.4 What steps do you take to interact with a deployed smart contract on the Ganache network using Web3.js?	3
4 Conclusion	4
References	5

1 Aim

The aim of this lab assignment is to understand the use of Ganache in blockchain development. Ganache is a personal blockchain for Ethereum development that you can use to deploy contracts, develop your applications, and run tests. This assignment will cover the setup and configuration of Ganache, deploying smart contracts using Truffle, and interacting with these contracts using Web3.js. By the end of this assignment, students will have a practical understanding of how to create, deploy, and test smart contracts in a local blockchain environment.

2 Demo

BLOCK	MINED ON	GAS USED	TRANSACTIONS
BLOCK 1	2022-01-05 19:04:03	21000	1 TRANSACTION
BLOCK 0	2022-01-05 18:40:33	0	NO TRANSACTIONS

Figure 1: Example of Ganache blocks

ADDRESS	BALANCE	TX COUNT	INDEX
0x627306090abaB3A6e1400e9345bC60c78a8BEf57	99.46 ETH	32	0
0xf17f52151EbEF6C7334FAD080c5704D77216b732	100.00 ETH	0	1
0xC5fdf4076b8F3A5357c5E395ab970B5B54098Fef	100.00 ETH	0	2
0x821aEa9a577a9b44299B9c15c88cf3087F3b5544	100.00 ETH	0	3

Figure 2: Ganache - Truffle Suite

3 Frequently Asked Questions

3.1 What is Ganache and how is it used in blockchain development?

Ganache is a personal Ethereum blockchain used for testing and development. It allows developers to deploy, test, and debug smart contracts in a controlled environment without interacting with the main Ethereum network. Ganache provides a local blockchain with features such as:

- Instant block mining, eliminating the need to wait for confirmations.

- Pre-funded accounts for easier testing.
- Detailed logging and debugging tools.
- Compatibility with Truffle Suite for streamlined smart contract deployment.

3.2 What are the key differences between Ganache test network and the real Ethereum network structure?

- **Consensus Mechanism:** Ganache does not use Proof of Work (PoW) or Proof of Stake (PoS); blocks are mined instantly for testing purposes.
- **Gas Fees:** Transactions on Ganache do not require real gas fees, while on Ethereum, gas fees fluctuate based on network demand.
- **Block Time:** Ganache mines blocks immediately, whereas Ethereum has an average block time of 12–15 seconds (PoS).
- **Network Security:** Ganache operates locally and is not connected to public nodes, while Ethereum relies on a decentralized global network.
- **Persistence:** Data on Ganache resets when restarted unless explicitly saved, whereas Ethereum maintains a permanent ledger.

3.3 How can you deploy a Solidity smart contract to a Ganache network using Truffle?

To deploy a Solidity smart contract on Ganache using Truffle, follow these steps:

1. Install Truffle and Ganache:

```
npm install -g truffle
```

2. Initialize a Truffle project:

```
truffle init
```

3. Compile the smart contract: Place the Solidity contract in the contracts/ folder and run:

```
truffle compile
```

4. Configure the Truffle migration script: Add a deployment script in the migrations/ directory.

5. Start Ganache: Run Ganache locally.

6. Deploy the contract: Run the migration command:

```
truffle migrate --network development
```

7. **Verify deployment:** Use Truffle console to interact with the deployed contract:

```
truffle console
```

3.4 What steps do you take to interact with a deployed smart contract on the Ganache network using Web3.js?

To interact with a deployed smart contract using Web3.js, follow these steps:

1. **Install Web3.js:**

```
npm install web3
```

2. **Connect to Ganache:**

```
const Web3 = require('web3');  
const web3 = new Web3('http://127.0.0.1:7545'); // Default Ganache RPC URL
```

3. **Load the smart contract ABI and address:**

```
const contractABI = [ /* ABI JSON */ ];  
const contractAddress = '0xYourContractAddress';  
const contract = new web3.eth.Contract(contractABI, contractAddress);
```

4. **Read contract data:**

```
contract.methods.getValue().call().then(console.log);
```

5. **Send a transaction:**

```
contract.methods.setValue(42).send({ from: '0xYourAccountAddress' });
```

6. **Listen for events:**

```
contract.events.ValueChanged({}, (error, event) => {  
  console.log(event);  
});
```

4 Conclusion

In conclusion, Ganache serves as an invaluable tool for blockchain developers by providing a personal Ethereum blockchain for testing and development purposes. It simplifies the process of deploying, testing, and debugging smart contracts without the need to interact with the main Ethereum network. By using Ganache in conjunction with tools like Truffle and Web3.js, developers can create robust and secure smart contracts, ensuring their functionality before deploying them to a live environment. This controlled and efficient development workflow ultimately contributes to the advancement and reliability of blockchain

References

- [1] Ganache - Truffle Suite. (n.d.). Retrieved from <https://www.trufflesuite.com/ganache>
- [2] Truffle Suite. (n.d.). Retrieved from <https://www.trufflesuite.com/>
Web3.js - Ethereum JavaScript API. (n.d.). Retrieved from <https://web3js.readthedocs.io/>
- [3] Ethereum. (n.d.). Retrieved from <https://ethereum.org/>
- [4] Solidity Documentation. (n.d.). Retrieved from <https://docs.soliditylang.org/>
- [5]