<https://www.linkedin.com/learning/building-an-ethereum-blockchain-app-4-ethereum-development-tools/installing-geth-lab?resume=false&u=2217001>

Geth

* Written in GO
* Full Ethereum node with access to blockchain
* Run on EVM

Mine ETH

Create Transactions and smart contracts

Examine existing blocks

. \Geth –synchmode “light

This runs geth as a light node on the Ethereum network

Ganache Lab – Test Blockchain to test smart contracts in isolated environment

Quickly fire up a personal Ethereum blockchain which you can use to run tests, execute commands, and inspect state while controlling how the chain operates.

# What is Truffle?[¶](https://archive.trufflesuite.com/docs/truffle/#what-is-truffle)

A world class development environment, testing framework and asset pipeline for blockchains using the Ethereum Virtual Machine (EVM), aiming to make life as a developer easier. With Truffle, you get:

* Built-in smart contract compilation, linking, deployment and binary management.
* [Advanced debugging](https://archive.trufflesuite.com/docs/truffle/how-to/debug-test/use-the-truffle-debugger) with breakpoints, variable analysis, and step functionality.
* Use [console.log](https://archive.trufflesuite.com/docs/truffle/reference/configuration/#soliditylog) in your smart contracts
* Deployments and transactions through MetaMask with [Truffle Dashboard](https://archive.trufflesuite.com/docs/truffle/how-to/use-the-truffle-dashboard) to protect your mnemonic.
* External script runner that executes scripts within a Truffle environment.
* Interactive console for direct contract communication.
* Automated contract testing for rapid development.
* Scriptable, extensible deployment & migrations framework.
* Network management for deploying to any number of public & private networks.
* Package management with NPM, using the [ERC190](https://github.com/ethereum/EIPs/issues/190) standard.
* Configurable build pipeline with support for tight integration.

<https://www.linkedin.com/learning/building-an-ethereum-blockchain-app-6-building-your-first-ethereum-app/writing-a-simple-smart-contract?resume=false&u=2217001>

// SPDX-License-Identifier: MIT

pragma solidity ^0.8.0; // vversion

// name of contract

*contract* HelloWorld {

    // helloMessage varialbe scope is only this contract

*string* *private* helloMessage = "HELLO WORLD";

    // public that anyone can use

    // view means it only deals with local variables and doesnt touch blockchain

*function* getHelloMessage() *public* *view* returns (*string* *memory*) {

        return helloMessage;

    }

}

truffle deploy –reset

The truffle deploy --reset command is used in the **Truffle framework**, which is a development environment for smart contracts on the Ethereum blockchain.

### Purpose:

* **Deployment of Smart Contracts**: Truffle deploys your smart contracts to a blockchain network (such as a local test network, a public testnet, or the Ethereum mainnet).
* **Reset Option (--reset)**: This option forces Truffle to redeploy all contracts, even if they haven't changed. Without this option, Truffle skips contracts that were already deployed based on a stored record of previous deployments. Using --reset clears this record and ensures that all contracts are redeployed from scratch.

### Use Case:

* This is particularly useful when you need to ensure that every contract is freshly deployed (for example, when testing different deployment scenarios, or when making sure that contracts are deployed in a clean state).