UNIT5 ETHEREUM WALLET & SMART CONTRACT **9Hrs.**

Ganache Full node Client – Intellij Plugin for Solidity – Truffle Suite: Create your Smart Contract – Connect Truffle to Smart Contract – Smart Contract: Hello world, MD5 Smart Contract, Smart Contract with truffle, Deploy the Smart Contract to your deployment network – Truffle Console – Operation with your Smart Contract via the Truffle CLI – Crypto currency Mining: Mining Hardware, Miner Types, Mining Pools, Mining Software

Ganache Full node Client

Overview

Forking and running a local simulated Ethereum environment is essential if you want to work with DeFi or do Ethereum development in general. In this guide, we'll cover how to fork Ethereum Blockchain with Ganache.

What is Ganache?

Ganache is an Ethereum developer tool that allows you to simulate a blockchain environment locally and test deployed smart contracts. You can use Ganache across the entire development cycle; enabling you to develop, deploy, and test your dApps in a safe and deterministic environment.

Smart contracts, once deployed on a blockchain, cannot be changed, so it becomes critical to thoroughly test and debug smart contracts before deploying them on the blockchain. Therefore it is essential to have a local blockchain environment that can free the developers from transaction costs and delays. Ganache is specifically designed for this. It is a local in-memory blockchain for development and testing purposes that simulate the real Ethereum network with some accounts funded with test Ether.

Why Fork Ethereum blockchain?

A fork in software development means making a copy of something separate from the original thing. Forking the Ethereum blockchain means copying the Ethereum blockchain's state at a certain block and making a copy of it to make your changes moving forward. This allows working with the Ethereum network without altering the actual Ethereum mainnet. The primary reason to do this is to work with existing smart contracts on the blockchain without recreating them. For example, if you want to work with a particular DeFi protocol, you can find that protocol's smart contract and fork the network for that block. This will allow you to test your project with the smart contract without making real transactions.

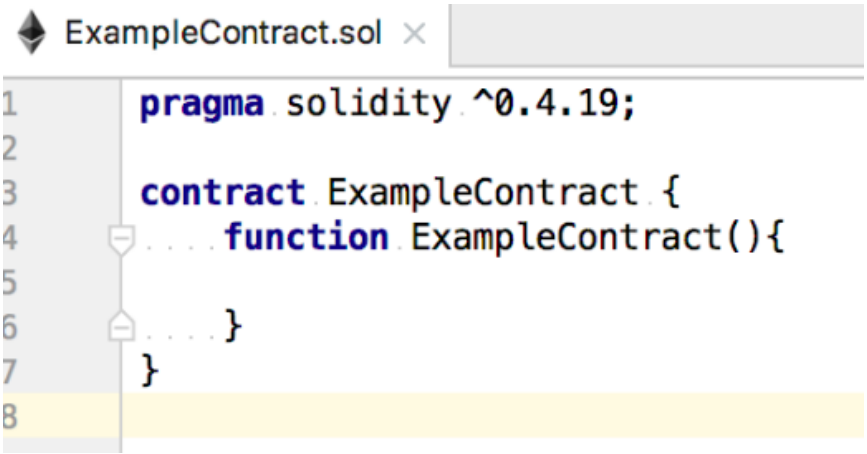
Booting our Ethereum node

We could use pretty much any Ethereum client, such as Geth or OpenEthereum (fka Parity for our purposes today. Since that is a bit too involved for forking a block, we'll just grab a

free endpoint from QuickNode to make this easy. We'll need a mainnet endpoint to get data from the chain as we are trying to make a simulated mainnet locally. After you've created your free ethereum endpoint, copy your HTTP Provider endpoint.

Intellij Plugin for Solidity

Intellij-Solidity—is a plugin for Intellij-based IDEs such as Webstorm, PhpStorm and PyCharm. This plugin offers syntax highlighting, code formatting and autocomplete for Solidity files. However, it doesn't provide any solutions for compiling and deploying contracts..



For compiling and deploying you can use Truffle Framework with Ganache CLI.

Truffle Setup

Firstly, you have to install Truffle globally using the npm package manager:

npm install -g truffle

Then you have to create a new directory for your project and go into it (the directory has to be empty):

mkdir MyFirstSolidityProject

cd MyFirstSolidityProject/

And finally run init truffle in our project directory:

truffle init

After initializing truffle in your project directory, you'll get the following files generated:

We will cover why we need these files in the next blog. For now, we'll just add some code to our truffle.js file to make it look like this:

```
module.exports = {
  networks: {
    development: {
      host: "127.0.0.1",
      port: 7545,
      network_id: "*"
    }
  }
}
```

Here we specified the environment, called development, running on 127.0.0.1:7545. We'll use it for deploying our contracts to the local Ethereum network.

Ganache CLI Setup

Now we have to install the Ganache CLI package globally with the npm package manager:

```
npm install -g ganache-cli
```

Ganache will run a local instance of the Ethereum network on your machine, which may be useful during development and debugging. We have to run ganache on the port specified for our development environment above (7545). This could be done by the following command:

```
ganache-cli -p 7545
```

After that you'll see a list of generated test account addresses, their private keys, and mnemonic phrases for the wallet:

Wrap Up

Now that we have the local network up and running, we can compile and migrate our project to that network. To do this, we have to run the following command:

```
truffle compile
```

This will generate a json file for our contracts under ./build/contracts directory. Then we will migrate this data to the network using the development environment:

truffle migrate --network development

Pros:

- Can be used in many modern IDE's without breaking previous settings
- Supports goto declarations, code completion, syntax highlighting
- Provides file templates for generated contracts and libraries
- Truffle and Ganache provide a big variety of project setups with React, Redux, Authentication etc.

Cons:

- The Intellij-Solidity plugin is currently in alpha, so it could have some bugs
- A lot of actions have to be done manually for setting up the compilation and deployment process

Truffle Suite:

Create your Smart Contract

Truffle is 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.

Truffle is widely considered the most popular tool for blockchain application development with over 1.5 million lifetime downloads. Truffle supports developers across the full lifecycle of their projects, whether they are looking to build on Ethereum, Hyperledger, Quorum, or one of an ever-growing list of other supported platforms. Paired with Ganache, a personal blockchain, and Drizzle, a front-end dApp development kit, the full Truffle suite of tools promises to be an end-to-end dApp development platform.

- Built-in smart contract compilation, linking, deployment and binary management.
- 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 EthPM & NPM, using the ERC190 standard.
- Interactive console for direct contract communication.
- Configurable build pipeline with support for tight integration.
- External script runner that executes scripts within a Truffle environment.

How it Works

Truffle Suite

Truffle is the most popular development tooling for Ethereum programmers. Easily deploy smart contracts and communicate with their underlying state without heavy client side programming. An especially useful library for the testing and iteration of Ethereum smart contracts.

Connecting with Truffle

Kaleido environments are provisioned with two pre-configured Truffle boxes for the easy construction and deployment of a basic decentralized application. Truffle is a particularly useful development framework, as it abstracts many of the web3 and Ethereum complexities that typically exist with blockchain app development and allows for rapid lightweight testing and iteration. This is a great way to get started without needing deep expertise around client libraries and JSON/RPC APIs.

The first box truffle-kaleido-box compiles and instantiates the simple storage smart contract. Simple storage uses a single global variable – storedData – specified as an unsigned integer and a single method – set – for state updates.

Why Kaleido

Kaleido's platform is different because it has everything businesses need to create complete blockchain solutions. With just a few clicks, you can create a blockchain network, deploy it globally, set up governance, and include additional services.

Blazing Fast Deployment, Speed, and Scale

- Deploy Production-Ready Blockchain Networks and Digital Assets in Minutes
- Amazingly Low Cost Per Transaction
- Multi-Party, Cross-Cloud, and Multi-Region Support
- Built-In High Availability and Disaster Recovery
- 400+ APIs and 40+ Services to Accelerate Development

No Lock-in and Open Source Technologies

- Support for Multiple Blockchain Protocols
- Enterprise Integrations and Marketplace
- Customizable Decentralization Options

- Actively Leading New Standards and Technologies

Proven Enterprise Platform and Expertise

- ISO and SOC2 Certified
- SLAs and 24/7 Support
- On-Chain and Off-Chain Services
- Secure Key Management
- Built-In Monitoring and Smart Contract Management

It is accompanied by the drizzle-kaleido-box which extends the initial library with a React and Redux front end for secure interaction with your Kaleido environment.

Smart Contract:

Hello world

STEP 1: CONNECT TO THE ETHEREUM NETWORK

There are many ways to make requests to the Ethereum chain. For simplicity, we'll use a free account on Alchemy, a blockchain developer platform and API that allows us to communicate with the Ethereum chain without having to run our own nodes.

STEP 2: CREATE YOUR APP (AND API KEY)

1. Navigate to the "Create App" page in your Alchemy Dashboard by hovering over "Apps" in the nav bar and clicking "Create App"

2. Name your app "Hello World", offer a short description, select "Staging" for the Environment (used for your app bookkeeping), and choose "Ropsten" for your network.

3. Click "Create app" and that's it! Your app should appear in the table below

STEP 3: CREATE AN ETHEREUM ACCOUNT (ADDRESS)

We need an Ethereum account to send and receive transactions. For this tutorial, we'll use MetaMask, a virtual wallet in the browser used to manage your Ethereum account address. More on transactions.

STEP 4: ADD ETHER FROM A FAUCET

In order to deploy our smart contract to the test network, we'll need some fake ETH. To get ETH you can go to the Ropsten faucet and enter your Ropsten account address, then click "Send Ropsten ETH." It may take some time to receive your fake ETH due to network traffic.

STEP 5: CHECK YOUR BALANCE

STEP 6: INITIALIZE OUR PROJECT

STEP 7: DOWNLOAD HARDHAT

STEP 8: CREATE HARDHAT PROJECT

STEP 9: ADD PROJECT FOLDERS

STEP 10: WRITE OUR CONTRACT

STEP 11: CONNECT METAMASK & ALCHEMY TO YOUR PROJECT

STEP 12: INSTALL ETHERS.JS

STEP 13: UPDATE HARDHAT.CONFIG.JS

STEP 14: COMPILE OUR CONTRACT

STEP 15: WRITE OUR DEPLOY SCRIPT

STEP 16: DEPLOY OUR CONTRACT

```
const HelloWorld = await ethers.getContractFactory("HelloWorld");
```

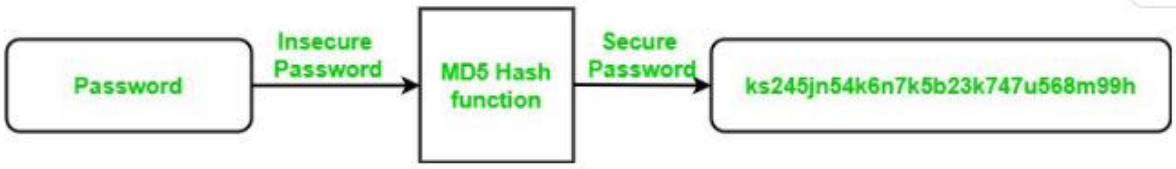
```
const hello_world = await HelloWorld.deploy();
```

MD5 Algorithm Smart Contract:

MD5 is a cryptographic hash function algorithm that takes the message as input of any length and changes it into a fixed-length message of 16 bytes. MD5 algorithm stands for the message-digest algorithm. MD5 was developed as an improvement of MD4, with advanced security purposes. The output of MD5 (Digest size) is always 128 bits. MD5 was developed in 1991 by Ronald Rivest.

Use Of MD5 Algorithm:

- It is used for file authentication.
- In a web application, it is used for security purposes. e.g. Secure password of users etc.
- Using this algorithm, We can store our password in 128 bits format.



MD5 Algorithm

Working of the MD5 Algorithm:

- MD5 algorithm follows the following steps

1. Append Padding Bits: In the first step, we add padding bits in the original message in such a way that the total length of the message is 64 bits less than the exact multiple of 512.

Suppose we are given a message of 1000 bits. Now we have to add padding bits to the original message. Here we will add 472 padding bits to the original message. After adding the padding bits the size of the original message/output of the first step will be 1472 i.e. 64 bits less than an exact multiple of 512 (i.e. 512*3 = 1536).

Length(original message + padding bits) = 512 * i – 64 where i = 1,2,3 . . .

2. Append Length Bits: In this step, we add the length bit in the output of the first step in such a way that the total number of the bits is the perfect multiple of 512. Simply, here we add the 64-bit as a length bit in the output of the first step.

i.e. output of first step = 512 * n – 64

length bits = 64.

After adding both we will get **512 * n** i.e. the exact multiple of 512.

3. Initialize MD buffer: Here, we use the 4 buffers i.e. J, K, L, and M. The size of each buffer is 32 bits.

- J = 0x67425301
- K = 0xEDFCBA45
- L = 0x98CBADFE
- M = 0x13DCE476

4. Process Each 512-bit Block: This is the most important step of the MD5 algorithm. Here, a total of 64 operations are performed in 4 rounds. In the 1st round, 16 operations will be performed, 2nd round 16 operations will be performed, 3rd round 16 operations will be performed, and in the 4th round, 16 operations will be performed. We apply a different function on each round i.e. for the 1st round we apply the F function, for the 2nd G function, 3rd for the H function, and 4th for the I function.

Application Of MD5 Algorithm:

- We use message digest to verify the integrity of files/ authenticates files.
- MD5 was used for data security and encryption.
- It is used to Digest the message of any size and also used for Password verification.
- For Game Boards and Graphics.

Advantages of MD5 Algorithm:

- MD5 is faster and simple to understand.
- MD5 algorithm generates a strong password in 16 bytes format. All developers like web developers etc use the MD5 algorithm to secure the password of users.
- To integrate the MD5 algorithm, relatively low memory is necessary.

- It is very easy and faster to generate a digest message of the original message.

Disadvantages of MD5 Algorithm:

- MD5 generates the same hash function for different inputs.
- MD5 provides poor security over SHA1.
- MD5 has been considered an insecure algorithm. So now we are using SHA256 instead of MD5
- MD5 is neither a symmetric nor asymmetric algorithm.

Crypto currency Mining:

Mining Hardware

Cryptocurrency Mining Hardware market report is expert study that can deliver you with an elaborate analysis of the Cryptocurrency Mining Hardware. The report covers information about top players, projected size of the market, data and figures to update about where opportunities are in the market, competitor analysis and vendor information. Also, it offers a complete analysis of the key market dynamics, with growth drivers, challenges, restraints, opportunities and trends. Furthermore, receive exact details and statistics associated to Cryptocurrency Mining Hardware market and its key factors such as revenue, growth,compound annual growth, year-over-year developments, consumption, and production.

The global Cryptocurrency Mining Hardware market size is estimated to be worth US$ 8680.3 million in 2022 and is forecast to a readjusted size of US$ 12930 million by 2028 with a CAGR of 6.9% during the review period.

Mining is the process of adding transaction records to Bitcoin's public ledger of past transactions (and a "mining rig" is a colloquial metaphor for a single computer system that performs the necessary computations for "mining". This ledger of past transactions is called the block chain as it is a chain of blocks. The blockchain serves to confirm transactions to the rest of the network as having taken place. Bitcoin nodes use the blockchain to distinguish legitimate Bitcoin transactions from attempts to re-spend coins that have already been spent elsewhere.

Competitive Landscape:

Report offers the Cryptocurrency Mining Hardware market competition landscape and a corresponding comprehensivestudy of the prominent players in this market, include

By Company

- BitMain Technologies Holding
- Canaan Creative
- Halong Mining
- Advanced Micro Devices
- Baikal Miner
- Bitfury Group
- Canaan Creative
- Innosilicon

- ASICMiner
- Ebang Communication

Market Segmentation:

Cryptocurrency Mining Hardware market report deliversstudy of the key trends in each sub-segment of the worldwideCryptocurrency Mining Hardware report, with estimates for development at the global, regional and country level and categorized the market based on product type, applications, regions.

Segment by Type

- ASIC Miner
- GPU Mining Rig
- Others

Segment by Application

- Enterprise
- Personal

Production by Region

- North America
- Europe
- China
- Japan
- South Korea
- Consumption by Region
- North America
- U.S.
- Canada
- Europe

TOC of Global Cryptocurrency Mining Hardware Market Insights and Forecast to 2028

1 Study Coverage

1.1 Cryptocurrency Mining Hardware Product Introduction

1.2 Market by Type

1.2.1 Global Cryptocurrency Mining Hardware Market Size by Type, 2017 VS 2021 VS 2028

Types of cryptocurrency mining

There are several types of cryptocurrency mining depending on the method you choose. Here are the most popular ways to mine Bitcoin.

ASIC mining

An application-specific integrated circuit (ASIC) is a specialized device built for one purpose, and ASIC miners are designed for mining a specific cryptocurrency. These are the most powerful option for Bitcoin mining. New ASICs can cost thousands of dollars, but they're also the only type of device where you can potentially make a profit from Bitcoin mining.

GPU mining

GPU mining uses one or more graphics cards to mine crypto. A typical "mining rig" is a computer that has one or more high-end graphics cards. This kind of mining is costly up front because you need to buy the graphics

cards. Although it's popular for mining other types of cryptocurrency, it doesn't work well for Bitcoin due to the lack of power compared to ASICs.

CPU mining

CPU mining uses a computer's central processing unit. This is the most accessible way to mine crypto since all you need is a computer, and it worked in the early days of Bitcoin. It's no longer recommended for mining Bitcoin because CPUs don't have nearly enough processing power to compete with ASICs.

Cloud mining

Cloud mining involves paying a company to mine crypto for you. Instead of setting up your own mining device, you're essentially renting one and receiving the profits after maintenance and electricity costs are deducted. While it may sound like a good deal at a glance, cloud mining normally requires committing to a contract, and, if crypto prices fall, you're unlikely to break even.

Mining pools

A mining pool is a group of crypto miners who pool their resources and share rewards. By working together, miners are much more likely to get the chance to mine new blocks. With Bitcoin mining, it's very difficult to mine blocks if you're operating solo. Each mining pool has its own hardware requirements, with most requiring you to have either an ASIC miner or a GPU.

What is Data Mining?

Data mining refers to digging into or mining the data in different ways to identify patterns and get more insights into them. It involves analyzing the discovered patterns to see how they can be used effectively.

In data mining, you sort large data sets, find the required patterns and establish relationships to perform data analysis. It’s one of the pivotal steps in data analytics, and without it, you can’t complete a data analysis process.

Data mining is among the initial steps in any data analysis process. Hence, it’s vital to perform data mining properly.

Learn data science courses from the World’s top Universities. Earn Executive PG Programs, Advanced Certificate Programs, or Masters Programs to fast-track your career.

What is Classification in Data Mining?

Classification in data mining is a common technique that separates data points into different classes. It allows you to organize data sets of all sorts, including complex and large datasets as well as small and simple ones.

It primarily involves using algorithms that you can easily modify to improve the data quality. This is a big reason why supervised learning is particularly common with classification in techniques in data mining. The primary goal of classification is to connect a variable of interest with the required variables. The variable of interest should be of qualitative type.

The algorithm establishes the link between the variables for prediction. The algorithm you use for classification in data mining is called the classifier, and observations you make through the same are called the instances. You use classification techniques in data mining when you have to work with qualitative variables.

There are multiple types of classification algorithms, each with its unique functionality and application. All of those algorithms are used to extract data from a dataset. Which application you use for a particular task depends on the goal of the task and the kind of data you need to extract

Types of Classification Techniques in Data Mining

Before we discuss the various classification algorithms in data mining, let's first look at the type of classification techniques available. Primarily, we can divide the classification algorithms into two categories:

1. Generative
2. Discriminative

Here's a brief explanation of these two categories:

Generative

A generative classification algorithm models the distribution of individual classes. It tries to learn the model which creates the data through estimation of distributions and assumptions of the model. You can use generative algorithms to predict unseen data.

A prominent generative algorithm is the Naive Bayes Classifier.

What Is a Mining Pool?

A mining pool is a joint group of cryptocurrency miners who combine their computational resources over a network to strengthen the probability of finding a block or otherwise successfully mining for cryptocurrency.

KEY TAKEAWAYS

- Cryptocurrency mining pools are groups of miners who share their computational resources.
- Mining pools utilize these combined resources to strengthen the probability of finding a block or otherwise successfully mining for cryptocurrency.
- If the mining pool is successful and receives a reward, that reward is divided among participants in the pool.

How a Mining Pool Works

Individually, participants in a mining pool contribute their processing power toward the effort of finding a block. If the pool is successful in these efforts, they receive a reward, typically in the form of the associated cryptocurrency.

Rewards are usually divided between the individuals who contributed, according to the proportion of each individual's processing power or work relative to the whole group. In some cases, individual miners must show proof of work in order to receive their rewards.

Rewards are usually split among the miners based on the agreed terms and on their respective contributions to the mining activity.

Anyone who wants to make a profit through cryptocurrency mining has the choice to either go solo with their own dedicated devices or to join a mining pool where multiple miners and their devices combine to enhance their hashing output. For example, attaching six mining devices that each offers 335 megahashes per second

(MH/s) can generate a cumulative 2 gigahashes of mining power, thereby leading to faster processing of the hash function.

Mining Pool Methods

Not all cryptocurrency mining pools function in the same way. There are, however, a number of common protocols that govern many of the most popular mining pools.

Proportional mining pools are among the most common. In this type of pool, miners contributing to the pool's processing power receive shares up until the point at which the pool succeeds in finding a block. After that, miners receive rewards proportional to the number of shares they hold.

Pay-per-share pools operate somewhat similarly in that each miner receives shares for their contribution. However, these pools provide instant payouts regardless of when the block is found. A miner contributing to this type of pool can exchange shares for a proportional payout at any time.

Peer-to-peer mining pools, meanwhile, aim to prevent the pool structure from becoming centralized. As such, they integrate a separate blockchain related to the pool itself and designed to prevent the operators of the pool from cheating as well as the pool itself from failing due to a single central issue.

Benefits of a Mining Pool

While success in individual mining grants complete ownership of the reward, the odds of achieving success is very low because of high power and resource requirements. Mining is often not a profitable venture for individuals. Many cryptocurrencies have become increasingly difficult to mine in recent years as the popularity of these digital currencies has grown and the costs associated with expensive hardware necessary to be a competitive miner as well as electricity oftentimes outweigh the potential rewards.

Mining pools require less of each individual participant in terms of hardware and electricity costs and increase the chances of profitability. Whereas an individual miner might stand little chance of successfully finding a block and receiving a mining reward, teaming up with others dramatically improves the success rate.

Disadvantages of a Mining Pool

By taking part in a mining pool, individuals give up some of their autonomy in the mining process. They are typically bound by terms set by the pool itself, which may dictate how the mining process is approached. They are also required to divide up any potential rewards, meaning that the share of profit is lower for an individual participating in a pool.

A small number of mining pools, such as AntPool, Poolin, and F2Pool dominate the bitcoin mining process, according to blockchain.com.1 Although many pools do make an effort to be decentralized, these groups consolidate much of the authority to govern the bitcoin protocol. For some cryptocurrency proponents, the presence of a small number of powerful mining pools goes against the decentralized structure inherent in bitcoin and other cryptocurrencies.