Chapter 3

Getting Your Hands on Blockchain

IN THIS CHAPTER

- **Creating and using a Bitcoin wallet**
- **Creating a simple smart contract**
- **Deploying a private blockchain**

Blockchains are very powerful tools and are positioned to change how the world moves money, secures systems, and builds digital identities. If you aren't a core developer, you probably won't be doing any in-depth blockchain development in the near future. That said, you still need to understand how blockchains work and what their core limitations are, because they'll be integrated into many everyday online interactions — from how businesses pay people to how governments know that their systems and data are intact and secure.

This chapter helps you get started in the blockchain world. You'll get acquainted with many of the most important aspects of working with blockchains and cryptocurrencies, yet you'll be working with tools that keep you at a comfortable distance from the intimidating and complex inner workings of blockchains. This chapter also helps you establish the basic crypto accounts that you need in later chapters.

Diving into the Bitcoin Blockchain

The Bitcoin blockchain is one of the largest and most powerful blockchains in the world. It was designed primarily to send Bitcoin, the cryptocurrency. So, naturally, in order to create a message in the Bitcoin blockchain, you must send some Bitcoins from one account to another.

When you send Bitcoins from one account to another, a transaction history is recorded in the Bitcoin blockchain. After a transaction has been entered, the information can't be removed — your message will be around as long as

Bitcoin is in existence. This concept of permanence is powerful — it's the most important characteristic of any blockchain.

You have several ways of adding extra little messages inside your transaction, but sometimes these methods don't always produce easily readable messages. In this section, I explain how to build the message directly into the Bitcoin transaction.

Embedding the data into the Bitcoin address ensures that it will be easily readable. You can do this by utilizing a Bitcoin vanity address. Think of a vanity address like a vanity license plate on a car. Six-letter Bitcoin vanity addresses can be obtained for free; longer ones cost money. The longer the vanity address, the more costly it is.

In this project, you create two Bitcoin wallets, add funds to one of them, obtain a vanity address, and send a little Bitcoin between your accounts.

TIP If you already have a Bitcoin wallet with funds in it, you can skip the first section and use that wallet.

Creating your first Bitcoin wallet

A Bitcoin wallet address is composed of 32 unique characters. It allows you to send and receive Bitcoins. Your private key is a secret code associated with your Bitcoin address that lets you prove your ownership of the Bitcoins linked with the address.

WARNING Anyone with your private key can spend your Bitcoins, so never share it.

Your first Bitcoin wallet needs to be linked to a credit card or bank account. I recommend using one of the following Bitcoin wallets:

- **Coinbase (** `www.coinbase.com` **)**
- **Xapo (** `www.circle.com` **)**

To set up your first wallet, just go to one of these URLs and create an account. It just takes a few minutes. When you have your account open, add a little money to it so you can experiment — $5 is a great starting point.

Creating a second Bitcoin wallet

To receive the Bitcoins you’ll send, you need to make a second Bitcoin wallet. For this second wallet, don’t use a Circle or Coinbase wallet — they don’t have the functionality you need for this purpose.

The easiest Bitcoin wallet to use for this project is the Blockchain.info wallet. Follow these steps to create it:

1. **Go to the Blockchain.info website (** www.blockchain.info **).**
2. **Click Wallet.**
3. **Click Create Your Wallet.**
4. **Enter an email address and password.**

Generating a Bitcoin vanity address

A Bitcoin vanity address is like having a personalized license plate for your car. It is a Bitcoin address that has a string of numbers or letters that appeals to you. A vanity address is optional, but a fun way to see your message in Bitcoin. There are a several free ways to create a Bitcoin wallet vanity address. My favorite is BitcoinVanityGen.com. To create your vanity address using BitcoinVanityGen.com, follow these steps:

1. **Go to the BitcoinVanityGen.com website (** www.bitcoinvanitygen.com **).**
2. **Enter six letters into the Type Letters field.**

   Bitcoin only allows for small messages, and your vanity address will make up the content of your message, which you can easily read in Bitcoin.

   TIP Choose something cool because you can reuse your address whenever you want after it has been created.

3. **Click Generate.**
4. **Click Email.**
5. **Enter your email address.**

   BitcoinVanityGen.com emails you when your vanity address has been found.
6. **Click the link in the email from BitcoinVanityGen.com.**

   You’ll be given your new vanity address and the private key associated with the address.
7. **Copy your address and private key, and keep them in a safe place.**

   You will need your address and private key for the next section.

WARNING Never share your private keys! Save your private key and a public key someplace safe. Use your public key to receiving or send Bitcoins. (You can share your public Bitcoin keys as much as you want.) The private key is the actual keys to your Bitcoins. If your private key is stolen or lost, you’ve lost your coins forever.

REMEMBER Cryptocurrency is unforgiving. Start off with small amounts of money when you’re learning how to use these systems.

Transferring your vanity address

In this section, you transfer your vanity address to a wallet. Transferring it will allow you to manage your address, and send and receive Bitcoins easily. Follow these steps to get started :

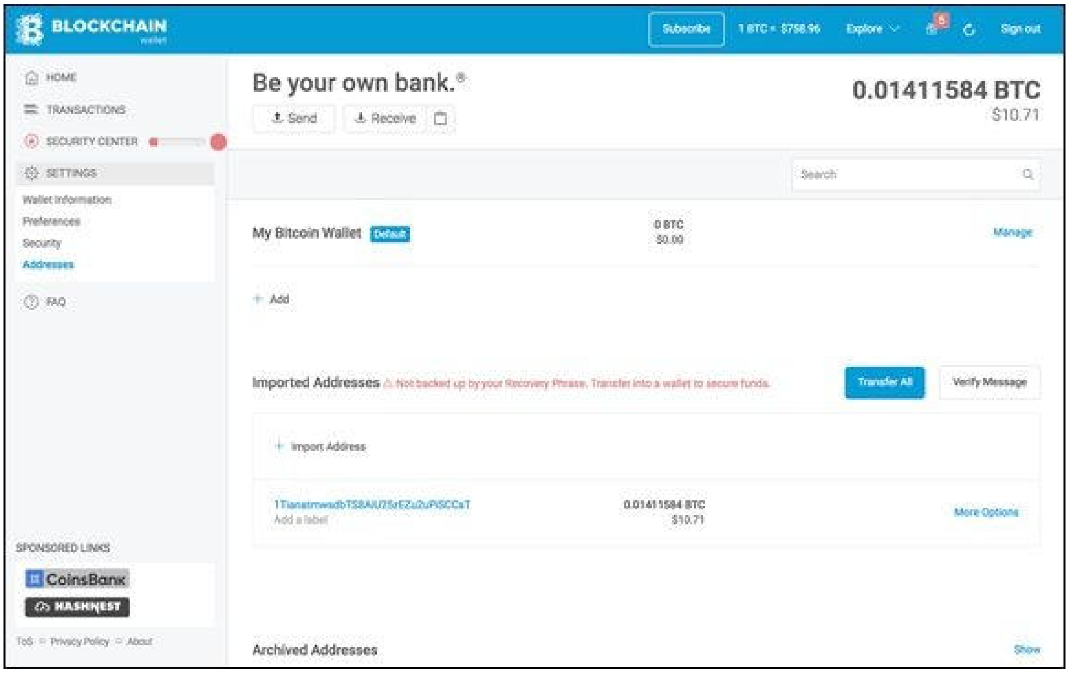
1. **Log into your Blockchain.info wallet (see “ Creating a second Bitcoin wallet ,” earlier in this chapter).**

   Figure 3-3 shows The settings page at blockchain.info.

2. **Click Settings and then click Addresses.**
3. **Next to Imported Addresses, click Manage Addresses.**

   The screen shown in Figure 3-1 appears.
4. **Click Import Address, enter your private key, and click Import.**

   You’ve now created an address that allows anyone to read your vanity address when you send or receive Bitcoins.



0.01411584 BTC
$10.71

My Bitcoin Wallet 0 BTC $0.00

Imported Addresses ⚠ Not backed up by your Recovery Phrase. Transfer into a wallet to secure funds.

1TianstmwsdbTS8AiU25rEZu2uPiSCCaT 0.01411584 BTC $10.71
Add a label

FIGURE 3-1: Managing your addresses.

Making an entry into the Bitcoin blockchain

Now that you have two Bitcoin wallets, you can make an entry into the Bitcoin blockchain. You do that by sending Bitcoins between your two wallets. Here’s how (the specifics vary from one wallet to the next, but this is the general idea):

1. **Log into the Bitcoin wallet that you added the initial funds to (see “ Creating your first Bitcoin wallet ,” earlier in this chapter).**

It prompts you to enter the recipient.

2. **Navigate to the page where you can send money, and copy and paste your vanity address (see "Generating a Bitcoin wallet vanity address ") into the address field.**
3. **Enter a small amount of money that you would like to send, and then click Send.**

Congratulations! You've just sent your first permanent message! You have forever engraved your message into the history of Bitcoin.

If you enjoyed learning how to do this and want to take your knowledge further, you can access a helpful online tutorial on sending Bitcoin messages at www.blockchainpie.com/blockchain-tutorial-bitcoin-message .

TECHNICAL STUFF

A Bitcoin transaction normally takes ten minutes to be confirmed, but could take several hours. The larger the value of the transaction, the longer you should wait. An unconfirmed transaction has not yet been included in the blockchain and is still reversible.

Reading a blockchain entry in Bitcoin

In the preceding section, I show you how to create a small permanent message in Bitcoin. Data on the Bitcoin blockchain is not encrypted because the data needs to be confirmed by the nodes. This means it will be easy to find the message that you created in the last project.

TIP

If you've just made the transfer of Bitcoins between your two wallets, wait about 10 or 15 minutes before following these steps.

1. **Go to the Blockchain.info website (www.blockchain.info).**
2. **Enter your vanity address in the Search box and press Enter.**

   The transaction page appears.

That’s all it takes to find your transaction and read the massage that you built into the address.

Using Smart Contracts with Bitcoin

A *smart contract* is autonomous software that can make financial decisions. The blockchain world is abuzz about smart contracts because they’re both amazing and terrifying in their implications for how the world economy operates.

In simple terms, a smart contract is a written contract that has been translated into code and build as complex if-then statements. The contract can self-verify that conditions have been met to execute the contract. It does this by pulling trusted data from outside sources. Smart contracts can also self-execute by releasing payment data or other types of data. They can be built around many different types of ideas and do not need to be financial in nature. Smart contracts can do all this while remaining tamper resistant from outside control.

Blockchain technology allowed smart contracts to come into existence because smart contracts offer the permanence and corrupt resistances that were once provided only by paper, ink, and a trusted authority to enforce it all. Smart contracts are a revolution in how we conduct business. They ensure that a contract will be executed as it was written. No outside enforcement is needed. The blockchain acts as the intermediary and enforcer.

Smart contracts are a big deal because when machines start executing contracts, it becomes difficult or impossible to undo. It also brings up an important nature of these instruments that can’t be overlooked and my first law of smart contracts: *She who controls the data, controls the contract.* All smart contracts verify an external data feed to prove performance and release payment to the correct party.

WARNING Although smart contracts are a revolutionary new technology, they can’t yet interpret the *intent* of the parties entering into the contract. Legal contracts in our society rely on people to interpret what the parties entering into the contract meant. Computers (at least so far) can only understand code, not the intent of the parties.

Building your first smart bond

A *smart bond* is a type of smart contract that can hold and release an object of value on its own, while also monitoring payments in various currencies using spot price data feeds. Many different types of smart contracts exist, and new ones are being invented every day.

Follow these steps to build your first smart bond:

1. **Go to the SmartContract website (`www.smartcontract.com`).**
2. **Click Sign Up.**

   The Sign Up page appears.
3. **Enter an email address and password and click Create an Account.**

   SmartContract sends you an email with a confirmation link.
4. **Click the link in the email sent to you by SmartContract to verify your account and log in.**
5. **Click Create Contract.**
6. **Click the Smart Bond tab (see Figure 3-2).**
7. **Click the Create Contract button.**

   You're ready to build your first if-then statement.
8. **Click the Smart Terms tab.**

   Smart contracts verify an outside data feed to prove the performance of your contract and trigger the release of payment. Here you choose the conditions that will trigger your smart contract.
9. **Choose Performance Monitoring.**

   Performance monitoring will look to see if an action has been taken outside of the contract. In your case, this will be the movement of funds from one account to another.
10. **In the If Payment To field, enter one of your Bitcoin addresses**

(created earlier in this chapter).

11. **In the Is field, enter a small dollar amount that you would like to transfer from one Bitcoin address to the other.**
12. **In the By Expiration Date field, enter a date a few days from now.**

    This sets the time parameters that the contract will use to monitor outside sources.
13. **Click the Description tab (see Figure 3-3).**
14. **In the Smart Contract Title field, enter a name for your contract.**
15. **In the Brief Description field, enter — you guessed it! — a brief description of the contract.**

    The description should act as a brief summary of the agreement’s purpose. Here you can also attach a legal document or other data, such as a image.
16. **Click the Attachments section.**

    WARNING Smart contracts are new technology and can have hiccups. It is best to only attach things that are unimportant and that you’d be okay exposing publicly.
17. **Click Attach Documents.**

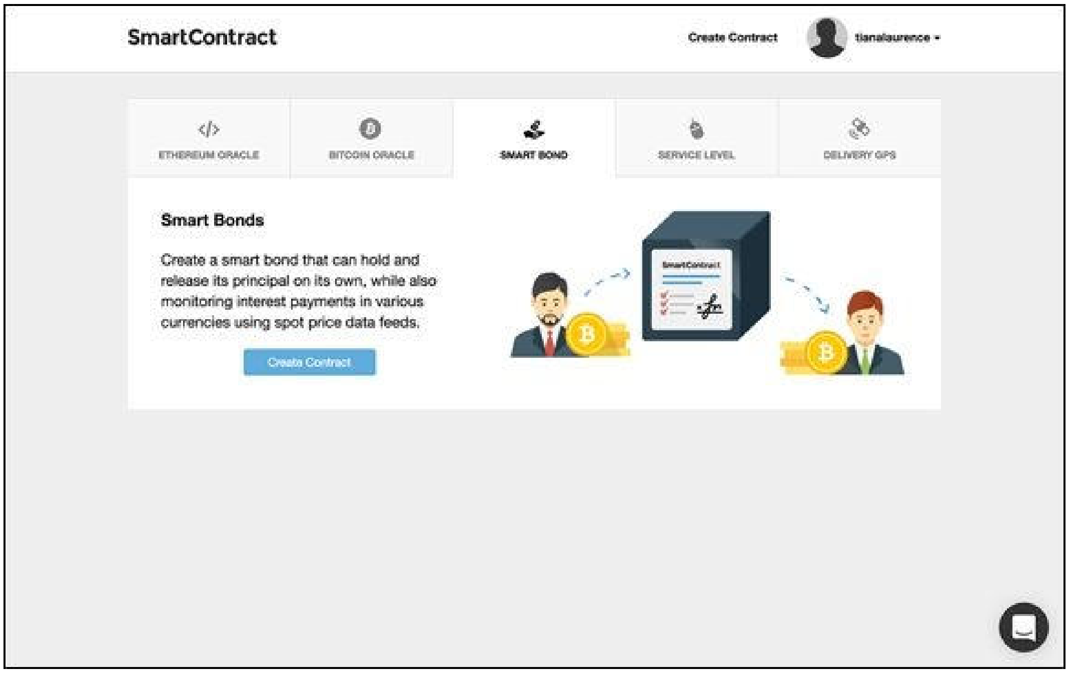
    You can attach an image or a PDF.
18. **Click the Sign & Send tab.**
19. **In the Address field, enter your email address to send yourself the contract.**
20. **Click the Finalize Contract button.**

    Now your contract will be monitoring the Bitcoin blockchain to monitor whether you send funds to the Bitcoin wallet address that you listed earlier.

21. **Return to your Bitcoin wallets and send funds between the two wallets.**

    Make sure to use the address and a little more than the amount that you listed on the contract in Steps 10 and 11. When the contract you created sees the record of the transaction of the Bitcoin blockchain, you'll be notified by email.

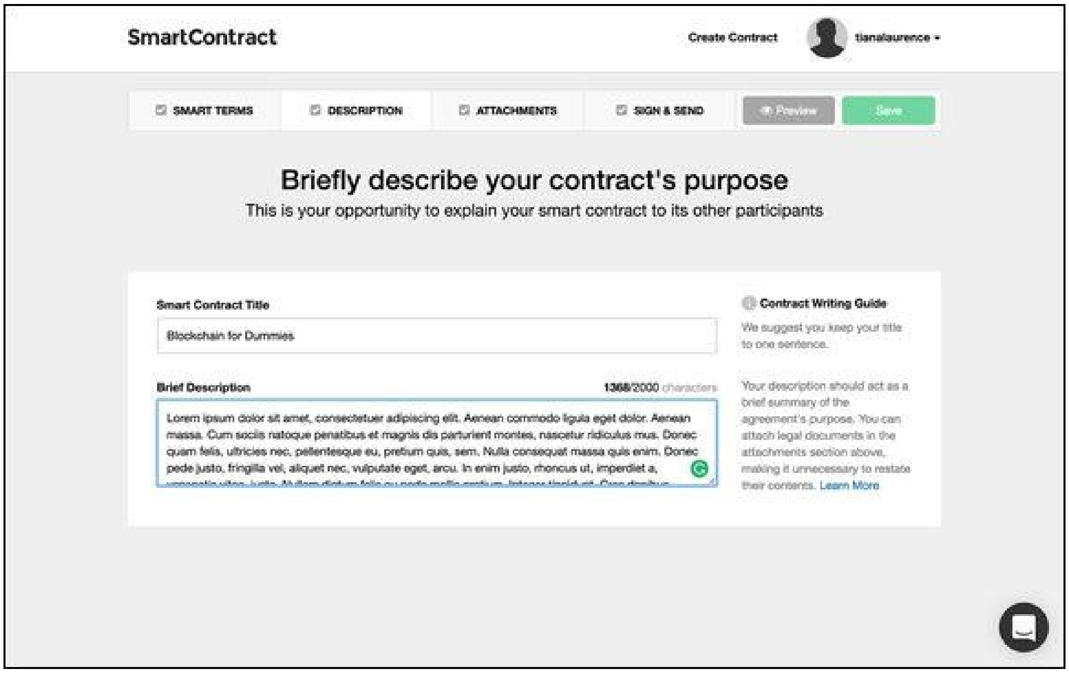
    REMEMBER The Bitcoin network will take a cut of the transaction, so add a little more to it so that it will meet the terms of the contract. For example, if you set the contract to $5, send $5.15 just to be safe.



Smart Bonds

Create a smart bond that can hold and release its principal on its own, while also monitoring interest payments in various currencies using spot price data feeds.

FIGURE 3-2: The Smart Bond tab.



Briefly describe your contract's purpose

This is your opportunity to explain your smart contract to its other participants

Smart Contract Title

Blockchain for Dummies

Brief Description — 1368/2000 characters

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim. Donec pede justo, fringilla vel, aliquet nec, vulputate eget, arcu. In enim justo, rhoncus ut, imperdiet a,

Contract Writing Guide

We suggest you keep your title to one sentence.

Your description should act as a brief summary of the agreement's purpose. You can attach legal documents in the attachments section above, making it unnecessary to restate their contents. Learn More

FIGURE 3-3: The Description tab.

Checking the status of your contract

You can check the status of your contract at any time by following these steps:

1. **Log into your SmartContract account at `www.smartcontract.com`.**
2. **Go to your Contract Dashboard.**

   After your transaction has been completed, the contract will show as complete. Your contract status is located below the Contract Dashboard.

REMEMBER Give the Bitcoin network 10 to 15 minutes to process your transaction before checking the status of it.

Building a Private Blockchain with

Docker and Ethereum

Private blockchains hold the promises of both having the benefits of a private database and the security of blockchains. The idea is most appealing for two reasons:

- **Private blockchains are great for developers because they allow them to test ideas without using cryptocurrency.** The developers' ideas can remain a secret as well, because the data has not been published publicly.
- **Large institutions can capitalize on the security and permanence of blockchain technology without their transactions being public the way they are in traditional blockchains.**

TIP

Most of this book assumes you're just learning about blockchain for the first time and have little to no programing skills, but this section requires some knowledge of GitHub, Docker, and how to use your computer's terminal. If you need a quick recap on coding before you dive in, I recommend *Coding For Dummies* by Nikhil Abraham (Wiley) for a great overview on coding for nontechnical people. If you don't plan to ever be hands-on with blockchain technology, you might want to skip the rest of this chapter.

In this section, you dive into building your first blockchain. You build it in two steps. The first step is to prepare your computer to create your private blockchain. Don't worry — it's made easier with tools from Docker and work that has been done by talented developers on GitHub. The second step is building your blockchain inside your Docker terminal.

Preparing your computer

You need to download some software on to your computer in order to try this blockchain project. Start by downloading the Docker Toolbox. Go to www.docker.com/toolbox to download the correct version for your operating system.

Next, download GitHub Desktop. Go to http://desktop.github.com . After you’ve installed GitHub Desktop, create a GitHub account at `www.github.com` by clicking Sign Up and entering a username, email address, and password, and then clicking the Sign Up for GitHub button.

Now you need to create a place to store your blockchain data. Create a folder on your computer’s desktop called `ethereum` . You’ll use this folder to hold your future repository and other files. Follow these steps to complete the process:

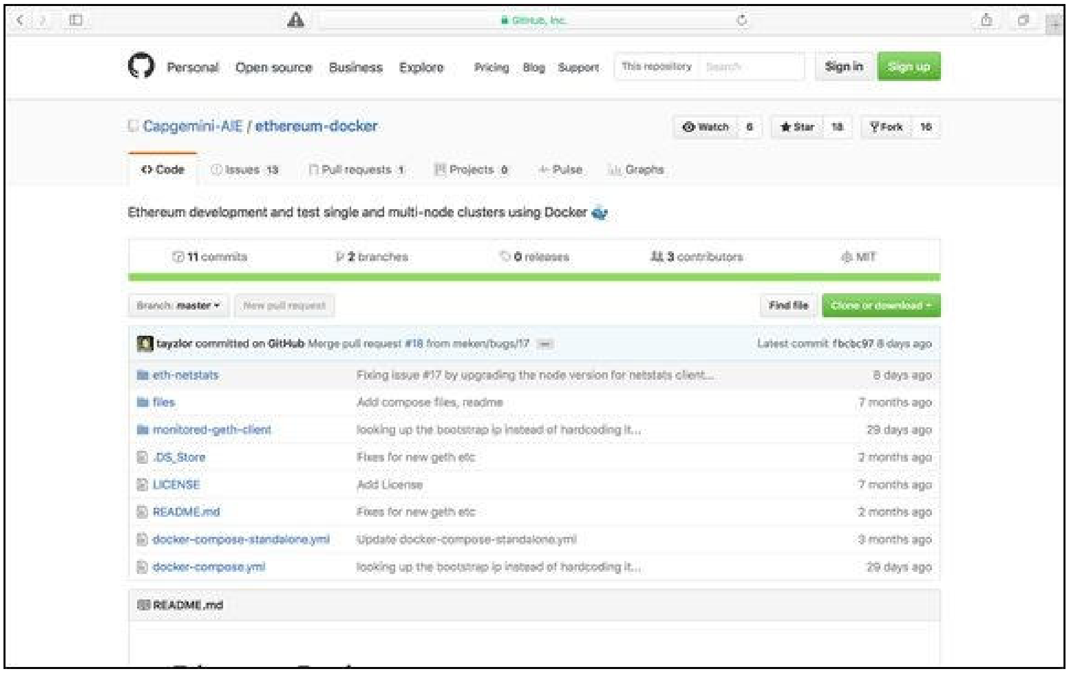
1. **Open GitHub Desktop.**
2. **Sign into the GitHub Desktop application on your computer with your new GitHub account.**
3. **Return to your web browser and go to `www.github.com/Capgemini-AIE/ethereum-docker`.**

   You see the page shown in Figure 3-4 .
4. **Click the Clone or Download button.**

   You’ll be given two choices: Open in Desktop or Download Zip (see Figure 3-5).
5. **Select the Open in Desktop option.**

   The GitHub Desktop application will reopens.

   In the GitHub Desktop application, navigate to the project folder `ethereum` and click Clone.



Capgemini-AIE / ethereum-docker

Ethereum development and test single and multi-node clusters using Docker

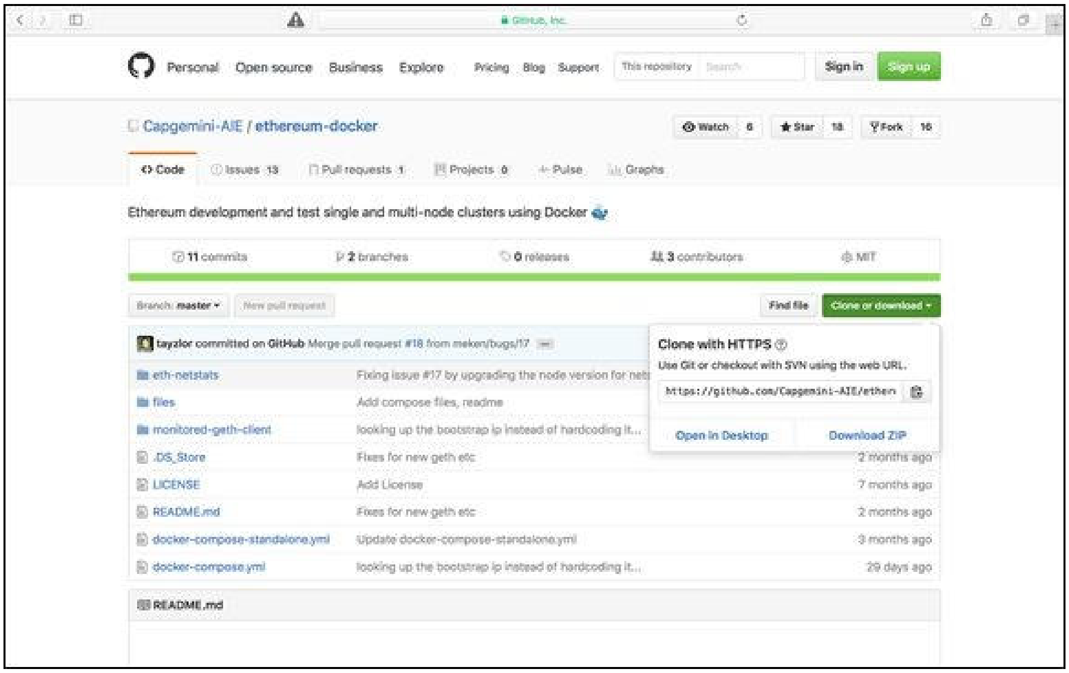
11 commits · 2 branches · 0 releases · 3 contributors · MIT

tayzlor committed on GitHub Merge pull request #18 from meken/bugs/17 … Latest commit fbcbc97 8 days ago

eth-netstats	Fixing issue #17 by upgrading the node version for netstats client…	8 days ago
files	Add compose files, readme	7 months ago
monitored-geth-client	looking up the bootstrap ip instead of hardcoding it…	29 days ago
.DS_Store	Fixes for new geth etc	2 months ago
LICENSE	Add License	7 months ago
README.md	Fixes for new geth etc	2 months ago
docker-compose-standalone.yml	Update docker-compose-standalone.yml	3 months ago
docker-compose.yml	looking up the bootstrap ip instead of hardcoding it…	29 days ago

README.md

FIGURE 3-4: Navigate to this page at GitHub.



Capgemini-AIE / ethereum-docker

Ethereum development and test single and multi-node clusters using Docker

11 commits · 2 branches · 0 releases · 3 contributors · MIT

tayzlor committed on GitHub Merge pull request #18 from meken/bugs/17 …

Clone with HTTPS

Use Git or checkout with SVN using the web URL.

https://github.com/Capgemini-AIE/ethere

eth-netstats	Fixing issue #17 by upgrading the node version for nets	
files	Add compose files, readme	
monitored-geth-client	looking up the bootstrap ip instead of hardcoding it…	
.DS_Store	Fixes for new geth etc	2 months ago
LICENSE	Add License	7 months ago
README.md	Fixes for new geth etc	2 months ago
docker-compose-standalone.yml	Update docker-compose-standalone.yml	3 months ago
docker-compose.yml	looking up the bootstrap ip instead of hardcoding it…	29 days ago

README.md

FIGURE 3-5: Open in Desktop.

Cloning from GitHub copies the information you need to build your new blockchain. Follow the steps in the next section to get started building your private blockchain.

Building your blockchain

You’re going to use the free Docker Quick Start Terminal tool to build your blockchain. It gives you access to a virtual machine, cutting down the time required to set up and debug your system. Because of these features, it lets you create a stable environment for your blockchain, so you don’t have to worry about the settings on your machine, and you can get up and running faster.

Follow these steps:

1. **Launch Docker on your computer using the Docker Quick Start Terminal.**

   TIP The Quick Start Terminal should be located with your applications or on your desktop.

   The Docker application launches a terminal you will use to build your blockchain.

2. **Change directories in the terminal to** `ethereum` **.**

   The files you create making the new blockchain will go into the desktop file you made in the preceding section. You need to give a command to the terminal in order to change directories. If you’re on a Mac or running Linux, enter the following command:

   ```
   cd ~ /Desktop/ethereum/ethereum-docker/
   ```

   If you’re on a PC, enter the following command:

   ```
   cd ~ \Desktop\ethereum\ethereum-docker\
   ```

   TIP If these commands don’t work for some reason, search the web for tutorials that explain how to change directories for your type of system.

   Now you can utilize the Ethereum–Docker files.

3. **Create one standalone Ethereum node by entering the following command into your terminal:**

   ```
   docker-compose -f docker-compose-standalone.yml up -d
   ```

   This one line of code will have created the following:

   - One Ethereum bootstrapped container
   - One Ethereum container that connects to the bootstrapped container
   - One Netstats container with a web UI to view activity in the cluster

4. **Take a look at your new blockchain by opening a web browser and going to** `http://$(docker-machine ip default):3000`**.**

Congratulations! You’ve built your own private blockchain. If you’re so inclined, say a word of thanks to Graham Taylor and Andrew Dong, who put a lot of time into creating the Ethereum–Docker integration.