

Projeto do Curso 1

Crie e transacione na blockchain privada Ethereum

Introdução

O curso 1 abordou os conceitos básicos de blockchain, estrutura, detalhes operacionais, algoritmos e técnicas.

Exploramos alguns conceitos de blockchain, como hashing e criptografia de chave pública, nos quizzes associados aos módulos 1, 2 e 3. Neste projeto de curso, exploraremos as seguintes operações em uma blockchain: criar nós em uma blockchain privada de Ethereum, criar contas, desbloquear contas, minerar, transacionar, transferir Ethers e verificar saldos.

Abordagem

Fornecemos uma imagem virtual (VM) que possui um cliente Ethereum pré-instalado (Go Language Ethereum – geth) e uma interface gráfica do usuário (GUI) para invocar os comandos necessários para realizar as tarefas do projeto.

Por favor, entenda que esta não é uma simulação. Você está, na verdade, trabalhando em nós Ethereum locais em seu laptop/dispositivo de computação. Utilizamos uma imagem virtual para que o projeto forneça as mesmas instruções, independentemente dos sistemas de computação em que nossos diversos alunos estejam trabalhando. Não é necessário o uso de comandos da interface de linha de comando (CLI), pois os comandos são codificados como botões. Este laboratório oferece acesso rápido e fácil aos comandos. Se você estiver curioso sobre a lista de comandos que usamos neste laboratório, ela está disponível no final da descrição deste laboratório.

Resultados da aprendizagem

Demonstre que você consegue seguir os passos para realizar transações em um blockchain, com os conceitos aprendidos no curso, como:

- Conta
- Saldo da conta
- Nó
- Pares
- Transação ponto a ponto
- Bloco de Gênesis
- Éteres
- Mineração
- Interação com um blockchain privado

Detalhes da implementação do projeto: O que fazer?

Neste projeto, você implantará um blockchain de teste Ethereum com dois nós e realizará transações entre eles.

Faremos isso em duas etapas: (1) configurando o ambiente e (2) trabalhando com o blockchain fornecido na Parte 1 e Parte 2.

Leia todas as instruções abaixo antes de começar o projeto.

1. Parte 1 – Configurando o ambiente

Consulte o arquivo de configuração do ambiente fornecido na Etapa 1 da aba Instruções. Siga as instruções para configurar e começar. NÃO pule esta etapa. Este documento também contém dicas de solução de problemas, caso você tenha algum problema.

(Você pode já ter feito isso antes de ler este documento)

2. Parte 2 – Inicializando e iniciando os nós Ethereum

Se você concluiu a etapa anterior com sucesso, verá uma interface web com instruções breves em cada etapa. Você pode concluir o exercício lendo as instruções fornecidas na interface web. **Após ler as notas abaixo, conclua todas as etapas indicadas.**

Observações:

• O que estamos fazendo na Parte 2?

- o Criação de dois nós
- o Conectando-os como pares
- o Verificando detalhes dos pares
- o Criação de múltiplas contas
- o Verificar se seus saldos iniciais são 0
- o Iniciar um minerador adiciona (taxas/recompensas de minerador) ao saldo da conta Coinbase do nó do minerador
- o Transação entre a conta com o saldo (primeiro nó) para QUALQUER outra conta no primeiro ou segundo nó.
- o Você também aprende a necessidade de desbloquear as contas antes das transações e a capacidade de examinar o status da transação (pendente, concluída, etc.)

Essas etapas são mostradas nas capturas de tela abaixo:

The screenshot shows a web interface titled "Getting Started with Ethereum". The first step, "1. Create Ethereum accounts", is highlighted. The instructions state: "In this first step, we need to create two new accounts for two of the Ethereum nodes. These 2 accounts will serve as the base accounts on their respective nodes. For creating your accounts, you'll need a password that you will use to lock or unlock your account when initiating a transaction. Please enter your password below for two accounts and click create accounts to get started." Below the text are two password input fields labeled "Enter your Password here for Node_1" and "Enter your Password here for Node_2". A blue "Create Accounts" button is positioned below the second field. Underneath, there are two sections for generated account addresses: "Generated Account address in Node_1" and "Generated Account address in Node_2", each with a corresponding text input field. The interface also shows links for subsequent steps: "2. Create the Genesis files", "3. Initialize a new Genesis Block for both the Nodes", and "4. Start the Ethereum Nodes". At the bottom, there is a "Move to Part 2" button.

Part 2: Operations on Ethereum

[← Return to Homepage](#)

Node_1

Enode

enode://88e27cd19b7ec015099fed317d9aa4ec131fbd302f5b6ea

Coinbase Address

0xee95c19f07ee5ec5d411b3183a7e54b0c3d18503

Node_2

Enode

enode://97522f0f7d5f25f8f5317078e25b402e9efa5e024556e78

Coinbase Address

0x88bd83f41b97e1adfb8eaf9d660652794f755b

Add Peer

We will connect the two nodes that you created to each other. To do so, we will add the enode of one of the nodes to the other and they both will be connected. Use the button below to connect them.

Connect Peers

Connected Peer Status

Add Peer Status

Check Peer Count

Check Peer Details

Create New Accounts

List Accounts and Check Balance

Miner

Unlock Account

Send Transaction

Check Transaction Status

Note: You can go to any of these above sections again in any order that you want to explore the tool.

Submit Project

Your submission token will be generated here.

Copy Token to Clipboard

- **Senhas para os nós**

Use senhas simples! (ex.: “b” para Nó1 e “c” para Nó2) para que você se lembre delas quando precisar desbloquear as contas.

Se você esquecer as senhas, terá que refazer os passos para criar as contas novamente.

É claro que, em um blockchain de produção, você terá que usar senhas fortes que outros não conseguirão adivinhar.

- **Leia as instruções e explicações em cada etapa da interface**

o Você precisa clicar nos botões em cada uma das etapas para executar as operações.

Aprenda à medida que avança.

Associe os passos que você está executando às aulas dos vídeos. A execução de cada operação leva algum tempo, dependendo dos recursos (como memória RAM) do seu computador.

o Seja paciente. **Preste atenção às notificações que solicitam que você espere até que cada a operação está concluída.**

Alocamos um tempo arbitrariamente longo/generoso para a conclusão de cada operação. Isso visa acomodar quaisquer variações no poder computacional dos alunos.

o Se sua interação parar por algum motivo, não hesite em voltar e reiniciar!

Há muitas opções para reiniciar (parcial ou totalmente do zero). A avaliação é baseada nas etapas que você conclui, e NÃO em quantas vezes você reiniciou.

(Consulte as dicas de solução de problemas caso tenha algum problema).

• Observações:

- o Valores Enode, o Endereço[0] ou endereços coinbase de contas
- o Detalhes do par no formato JSON
- o Criação de contas em um único nó
- o Operações de mineração
- o O bloco de gênese é o primeiro bloco de uma blockchain
- o Geração de DAG (gráfico acíclico direcionado) quando um minerador é iniciado
 - ÿ O processo é necessário para memória com base no consenso da Prova de Trabalho no Ethereum.

• Iniciando e Parando o Minerador

Neste projeto, há apenas um minerador iniciado no primeiro nó. Não interrompa o minerador até concluir as transações. Lembre-se da lição sobre a necessidade de um minerador para confirmar suas transações. As transações nunca serão confirmadas se você não tiver um minerador em execução.

• Classificação e Submissão

- o Estamos usando classificação automática para seu projeto.
 - o Você é avaliado pelas etapas concluídas.
 - o A nota NÃO depende do número de tentativas.
 - o A nota NÃO depende do tempo que você levou para concluir o projeto.
- Ao concluir o projeto, você receberá um hash exclusivo com base nas etapas concluídas. Envie-o ao sistema de aprendizagem do Coursera conforme as instruções para obter uma nota.

o Salve o hash gerado em um arquivo de texto (.txt) e envie este arquivo.

o Você receberá um erro se simplesmente enviar o hash.

o Veja este link se precisar de ajuda: [https://learner.coursera.help/hc/en-](https://learner.coursera.help/hc/en-nós/artigos/209818753)

[nós/artigos/209818753](https://learner.coursera.help/hc/en-nós/artigos/209818753)

• Explorando Além

Este projeto está sempre disponível para você explorar e aprender. Não hesite em explorar mais e experimentar diferentes transações. Fornecemos apenas um minerador. Você pode reiniciar a VM e experimentar diversas operações, incluindo os comandos CLI (interface de linha de comando) fornecidos no final deste documento. A mesma VM será usada no Curso 3 para o desenvolvimento de uma Aplicação Descentralizada ou Dapp.

• Resumo

Neste exercício, você teve a ideia de configurar seu próprio cliente Ethereum e trabalhar em algumas operações básicas.

Recomendamos fortemente que você use as Discussões Online para discutir o projeto com outros alunos e resolver quaisquer problemas que você possa enfrentar.

Nos próximos dois cursos, você projetará e desenvolverá aplicativos para o blockchain Ethereum.

Material extra

Neste projeto, já instalamos o **geth** no dispositivo virtual fornecido a você. Para facilitar, disponibilizamos uma interface web para ajudar você a interagir com a instância do geth em execução no backend. Se tiver curiosidade, aqui estão os comandos por trás dos botões que você clicou. Aprenderemos mais detalhes sobre os comandos geth e as APIs de suporte no Curso 3.

1. Crie novas contas:

geth conta nova --datadir ~/Node_1 e geth conta nova --datadir ~/Node_2

2. Inicialize o arquivo Genesis:

geth init customGenesis.json --datadir ~/Node_1 e geth init customGenesis.json --datadir ~/Node_2

3. Inicie o Geth:

a. NÓ_1:

```
geth --datadir ~/Node_1 --maxpeers 95 --networkid 13 --nodiscover --rpc --  
"*" --port 30301 --rpcport 8544 --  
rpcapi="txpool,db,eth,net,web3,pessoal,admin,miner"
```

b. NÓ_2:

```
geth --datadir ~/Node_2 --maxpeers 95 --networkid 13 --nodiscover --rpc --  
"*" --port 30302 --rpcport 8545 --  
rpcapi="txpool,db,eth,net,web3,pessoal,admin,miner"
```

4. **Conectar pares:** admin.addPeer(<eNode do outro nó>);

5. **Contagem de pares:** net.peerCount

6. **Detalhes do par:** admin.peers

7. **Crie novas contas:** personal.newAccount(<senha>)

8. **Listar contas:** eth.accounts

9. **Verificar saldo:** eth.getBalance(<conta>)

10. **Inicie o Miner:** miner.start(4)

11. **Pare o minerador:** miner.stop()

12. **Desbloquear contas:** personal.unlockAccount(<endereço>,<senha>,<duração em segundos>)

13. **Enviar transação:** eth.sendTransaction({from:<address>, to:<address> 14. , valor: <valor>})

Verificar status da transação: txpool.status