

PROJECT 15
MAKING AN ETHEREUM CONTRACT
WITH TRUFFLE
SPRING TRIMESTER 2022
By
NIKHIL PATEL

OVERVIEW : In this lab, We will learn how to install truffle prerequisites such as node, nvm, and npm in this lab. We will develop an Ethereum smart contract in this part. Ethereum's capacity to do these sorts of applications separates it from Bitcoin.

- Truffle is used to run smart contracts written in Solidity, and it may also be tested without utilizing actual ether. It is simple to operate a truffle via truffle init, truffle develop, and truffle start. Smart contracts are simple to install on a network.
- The truffle migrate command is in charge of migrating javascript, which allows you to deploy contracts on the network. In truffle, we can construct a smart contract (create contract proofexistence1), which will automatically produce the files needed for the smart contract.

PROCEDURE :

1) Installing Node.js and npm on Ubuntu 16.04

Go to terminal and execute the following :

```
sudo apt install -y curl
```

When you get this error Could not get lock /var/lib/dpkg/lock-frontend - open

To remove the file/folder execute:

```
sudo rm /var/lib/dpkg/lock-frontend
```

```
sudo apt install -y curl
```

In the same terminal execute this command:

```
curl -fsSL https://deb.nodesource.com/setup_16.x | sudo -E bash -
```

```
sudo apt-get install -y nodejs
```

Open a terminal, execute :

```
node -v
```

```
npm -v
```

```
sudo npm install npm@latest -g
```

A screenshot of a Kubuntu desktop environment. In the center is a terminal window titled "bash — Konsole". The terminal shows a series of commands being run on a Kubuntu system:

```
kubuntu@kubuntu-virtual-machine:~$ sudo apt update  
sudo apt install curl git make build-essential -y  
curl -sL https://deb.nodesource.com/setup_8.x | sudo bash -  
sudo apt install nodejs -y  
node -v  
npm -v  
[sudo] password for kubuntu:  
Get:1 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]  
Hit:2 https://dl.yarnpkg.com/debian stable InRelease  
Get:3 https://deb.nodesource.com/node_16.x jammy InRelease [4,583 B]  
Get:4 http://security.ubuntu.com/ubuntu jammy-security/main i386 Packages [77.9 kB]  
Get:5 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [223 kB]  
Get:6 http://security.ubuntu.com/ubuntu jammy-security/main Translation-en [54.3 kB]  
Get:7 http://security.ubuntu.com/ubuntu jammy-security/main amd64 DEP-11 Metadata [11.4 kB]  
Get:8 http://security.ubuntu.com/ubuntu jammy-security/main amd64 c-n-f Metadata [3,564 B]  
Get:9 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [200 kB]  
Get:10 http://security.ubuntu.com/ubuntu jammy-security/restricted i386 Packages [21.8 kB]  
Get:11 http://security.ubuntu.com/ubuntu jammy-security/restricted Translation-en [30.0 kB]  
Get:12 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 c-n-f Metadata [512 B]  
Get:13 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [93.5 kB]
```

The terminal window has a standard KDE-style menu bar with "File", "Edit", "View", "Bookmarks", "Plugins", "Settings", and "Help". The title bar says "Proj 15 Ethereum with Truffle_2022". The desktop background is orange and blue abstract. The bottom of the screen shows the Kubuntu taskbar with icons for the terminal, file manager, and browser, along with a system tray showing the date and time (2:28 PM, 7/14/22).

Proj 15 Ethereum with Truffle_2022

```
~ : bash — Konsole
File Edit View Bookmarks Plugins Settings Help
Hit:6 http://security.ubuntu.com/ubuntu jammy-security InRelease
Reading package lists... Done

## Confirming "jammy" is supported...

+ curl -sLf -o /dev/null 'https://deb.nodesource.com/node_8.x/dists/jammy/Releas
e'

## Your distribution, identified as "jammy", is not currently supported, please
contact NodeSource at https://github.com/nodesource/distributions/issues if you
think this is incorrect or would like your distribution to be considered for sup
port

Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages will be upgraded:
  nodejs
1 upgraded, 0 newly installed, 0 to remove and 105 not upgraded.
Need to get 26.5 MB of archives.
After this operation, 6,144 B of additional disk space will be used.
Get:1 https://deb.nodesource.com/node_16.x jammy/main amd64 nodejs amd64 16.16.0
-deb-1nodesource1 [26.5 MB]
Fetched 26.5 MB in 9s (2,863 kB/s)
(Reading database ... 214590 files and directories currently installed.)
Preparing to unpack .../nodejs_16.16.0-deb-1nodesource1_amd64.deb ...
Unpacking nodejs (16.16.0-deb-1nodesource1) over (16.15.1-deb-1nodesource1) ...
Setting up nodejs (16.16.0-deb-1nodesource1) ...
Processing triggers for man-db (2.10.2-1) ...
v16.15.1
Unknown command: "-v"

To see a list of supported npm commands, run:
  npm help
kubuntu@kubuntu-virtual-machine:~$
```

2) Installing Truffle

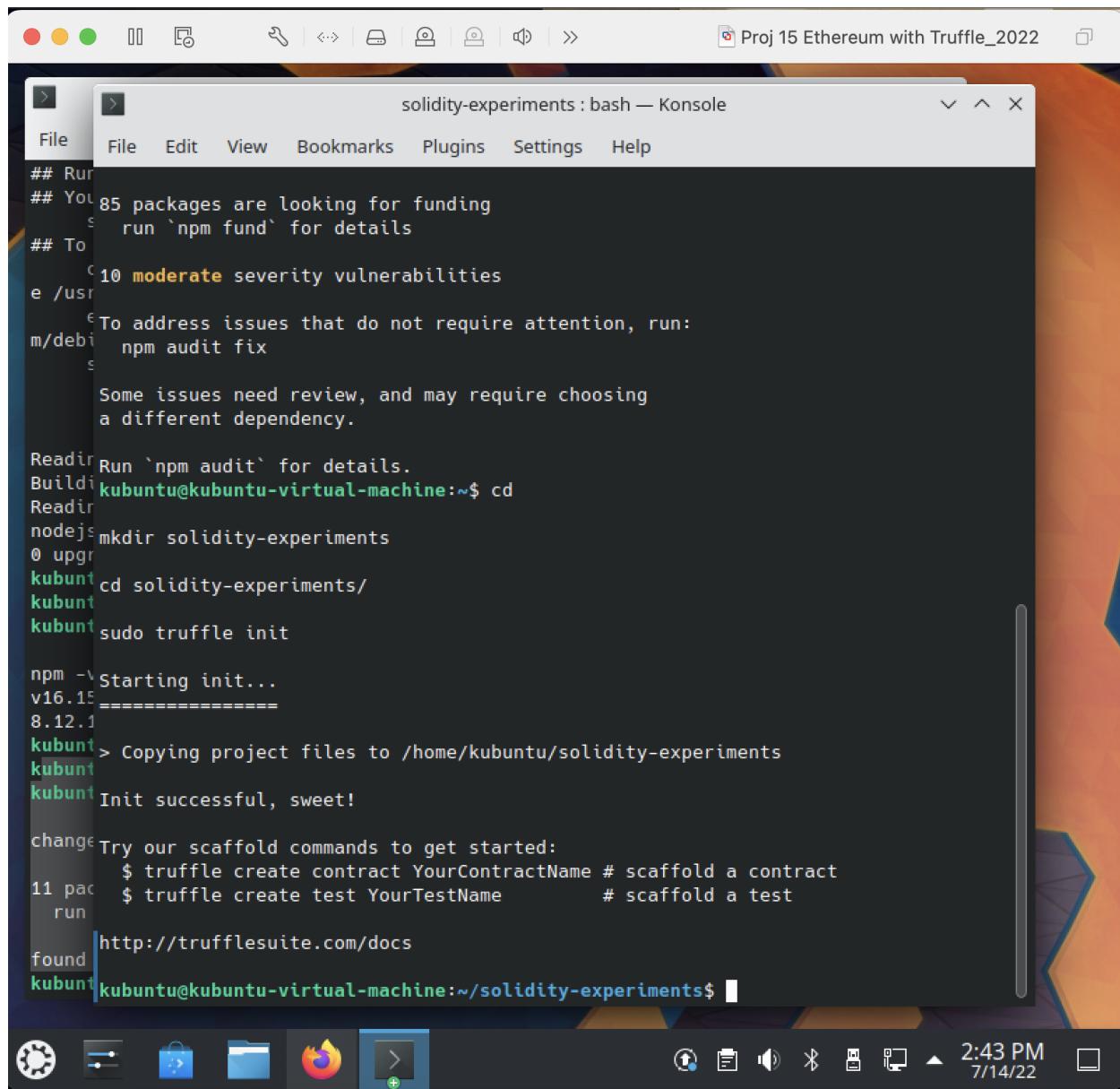
Open a New Terminal, execute :

```
sudo npm install -g truffle
```

3) Set Up a Project

In the terminal execute the following command:

```
cd  
mkdir solidity-experiments  
cd solidity-experiments/  
sudo truffle init
```

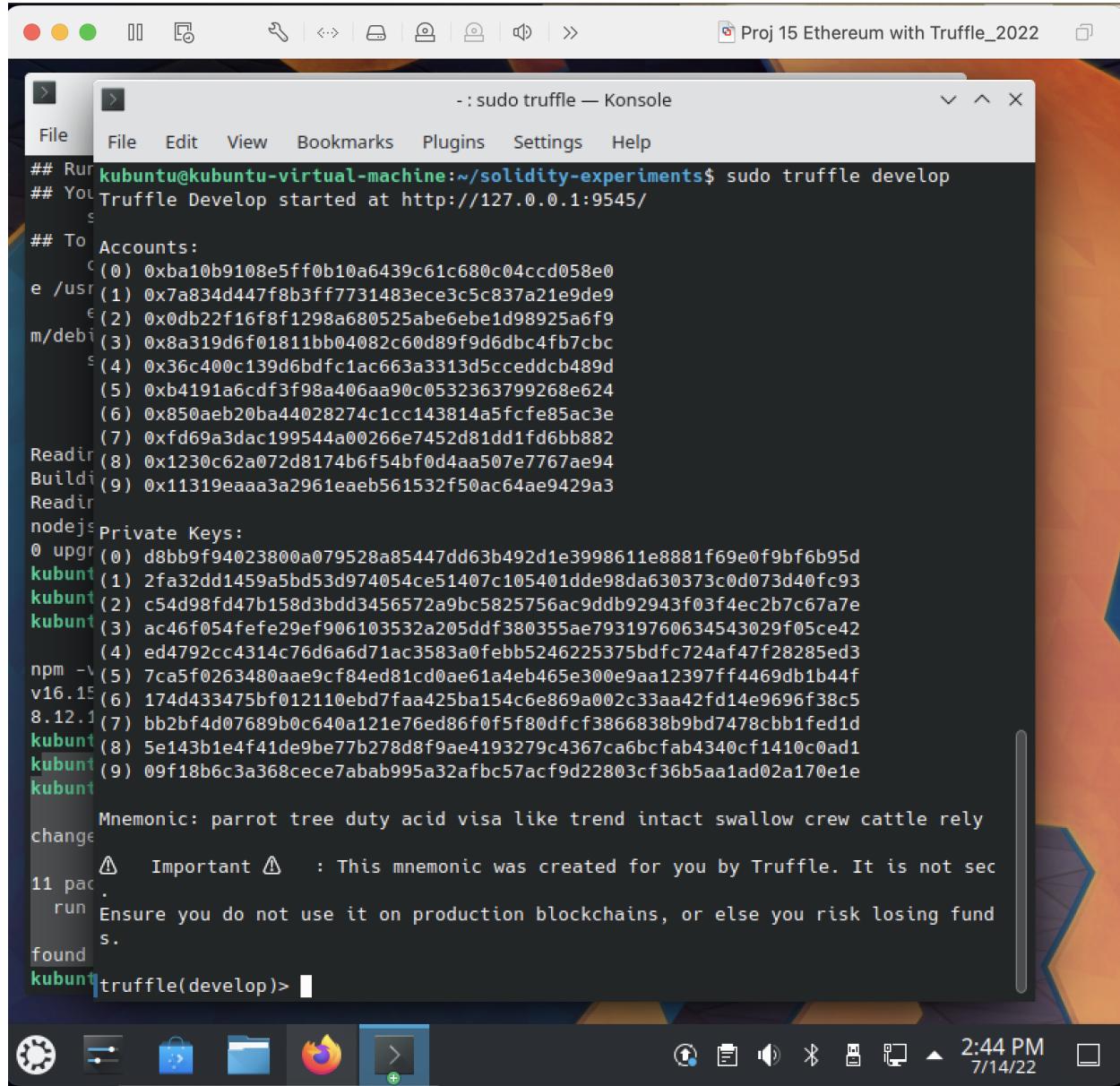


```
solidity-experiments : bash — Konsole  
File File Edit View Bookmarks Plugins Settings Help  
## Run  
## You 85 packages are looking for funding  
## To S run `npm fund` for details  
## To C 10 moderate severity vulnerabilities  
e /usr  
e To address issues that do not require attention, run:  
m/debi m npm audit fix  
S Some issues need review, and may require choosing  
a different dependency.  
Readir Run `npm audit` for details.  
Buildi kubuntu@kubuntu-virtual-machine:~$ cd  
Readir  
nodejs mkdir solidity-experiments  
0 upgr  
kubunt cd solidity-experiments/  
kubunt  
kubunt sudo truffle init  
npm -v Starting init...  
v16.15 =====  
8.12.1  
kubunt > Copying project files to /home/kubuntu/solidity-experiments  
kubunt  
kubunt Init successful, sweet!  
change Try our scaffold commands to get started:  
      $ truffle create contract YourContractName # scaffold a contract  
11 pac  $ truffle create test YourTestName      # scaffold a test  
run  
found http://trufflesuite.com/docs  
kubunt kubuntu@kubuntu-virtual-machine:~/solidity-experiments$
```

4) Starting the Development Testchain

In the terminal execute the following command:

```
sudo truffle develop
```



```
- : sudo truffle — Konsole
File   File   Edit   View   Bookmarks   Plugins   Settings   Help
## Run kubuntu@kubuntu-virtual-machine:~/solidity-experiments$ sudo truffle develop
## You Truffle Develop started at http://127.0.0.1:9545/
## To Accounts:
e /usr (0) 0xba10b9108e5ff0b10a6439c61c680c04cc058e0
m/debi (1) 0x7a834d447f8b3ff7731483ece3c5c837a21e9de9
(2) 0x0db22f16f8f1298a680525abe6ebe1d98925a6f9
(3) 0x8a319d6f01811bb04082c60d89f9d6dbc4fb7cbc
(4) 0x36c400c139d6bdfc1ac663a3313d5cceddcba89d
(5) 0xb4191a6cdf3f98a406aa90c0532363799268e624
(6) 0x850aeb20ba44028274c1cc143814a5fcfe85ac3e
(7) 0xfd69a3dac199544a00266e7452d81dd1fd6bb882
Readin (8) 0x1230c62a072d8174b6f54bf0d4aa507e7767ae94
Buildin (9) 0x11319aaaa3a2961eaeb561532f50ac64ae9429a3
Readin
nodejs Private Keys:
0 upgr (0) d8bb9f94023800a079528a85447dd63b492d1e3998611e8881f69e0f9bf6b95d
kubunt (1) 2fa32dd1459a5bd53d974054ce51407c105401dde98da630373c0d073d40fc93
kubunt (2) c54d98fd47b158d3bdd3456572a9bc5825756ac9ddb92943f03f4ec2b7c67a7e
kubunt (3) ac46f054fefe29ef906103532a205ddf380355ae79319760634543029f05ce42
(4) ed4792cc4314c76d6a6d71ac3583a0febb5246225375bdfc724af47f28285ed3
npm ~\ (5) 7ca5f0263480aae9cf84ed81cd0ae61a4eb465e300e9aa12397ff4469db1b44f
v16.15 (6) 174d433475bf012110ebd7faa425ba154c6e869a002c33aa42fd14e9696f38c5
8.12.1 (7) bb2bf4d07689b0c640a121e76ed86f0f5f80dfcf3866838b9bd7478ccb1fed1d
kubunt (8) 5e143b1e4f41de9be77b278d8f9ae4193279c4367ca6bcfab4340cf1410c0ad1
kubunt (9) 09f18b6c3a368cece7abab995a32afbc57acf9d22803cf36b5aa1ad02a170e1e
kubunt

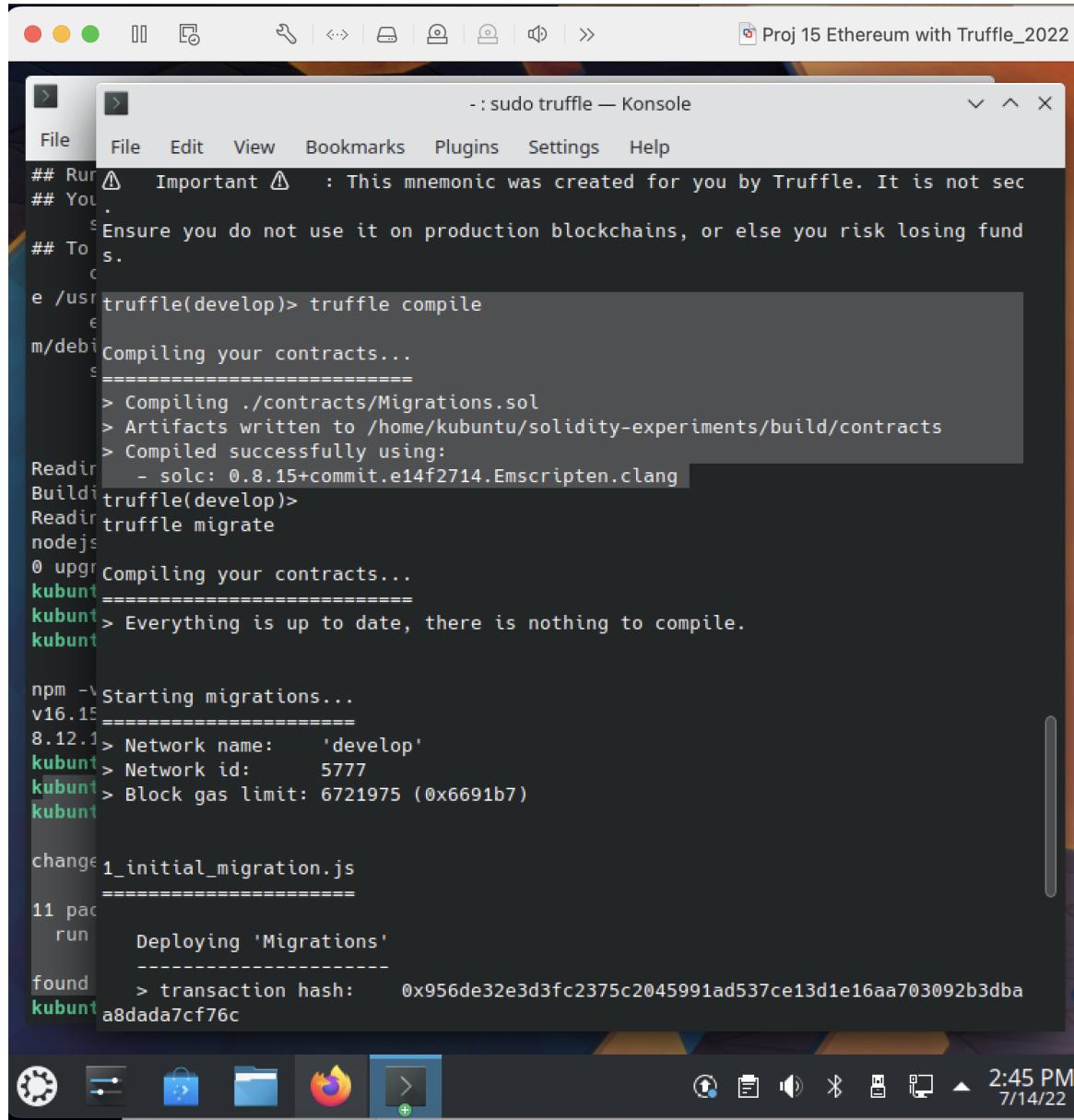
Mnemonic: parrot tree duty acid visa like trend intact swallow crew cattle rely
change
⚠️ Important ⚠️ : This mnemonic was created for you by Truffle. It is not sec
11 pac
run Ensure you do not use it on production blockchains, or else you risk losing fund
s.
found
kubunt|truffle(develop)> █
```

5) Compiling and Deploying Sample Contracts

In the terminal, at the **truffle(develop)>** prompt execute :

truffle compile

truffle migrate



```
- : sudo truffle — Konsole
File File Edit View Bookmarks Plugins Settings Help
## Run ⚠️ Important ⚠️ : This mnemonic was created for you by Truffle. It is not sec
## You . Ensure you do not use it on production blockchains, or else you risk losing fund
## To s.
e /usr/truffle(develop)> truffle compile
m/debiCompiling your contracts...
=====
> Compiling ./contracts/Migrations.sol
> Artifacts written to /home/kubuntu/solidity-experiments/build/contracts
> Compiled successfully using:
Readin - solc: 0.8.15+commit.e14f2714.Emscripten.clang
Buildin truffle(develop)>
Readin truffle migrate
nodejs
0 upgrCompiling your contracts...
kubunt =====
kubunt > Everything is up to date, there is nothing to compile.
kubunt

npm ~\Starting migrations...
v16.15=====
8.12.1> Network name: 'develop'
kubunt > Network id: 5777
kubunt > Block gas limit: 6721975 (0x6691b7)
kubunt

change 1_initial_migration.js
=====
11 pac run Deploying 'Migrations'
found > transaction hash: 0x956de32e3d3fc2375c2045991ad537ce13d1e16aa703092b3dba
kubunt a8dada7cf76c
```

- : sudo truffle — Konsole

```
File Edit View Bookmarks Plugins Settings Help
## Run - solc: 0.8.15+commit.e14f2714.Emscripten.clang
## You truffle(develop)>
$ truffle migrate
## To
` Compiling your contracts...
e /usr=====
` Everything is up to date, there is nothing to compile.
m/debi
` Starting migrations...
=====
` Network name:      'develop'
Readir` Network id:      5777
Buildi` Block gas limit: 6721975 (0x6691b7)
Readir
nodejs
0 upgr1_initial_migration.js
kubunt=====
kubunt
kubunt Deploying 'Migrations'
-----
` transaction hash: 0x956de32e3d3fc2375c2045991ad537ce13d1e16aa703092b3dba
a8dada7cf76c
8.12.1` Blocks: 0           Seconds: 0
kubunt` contract address: 0x91a2A73D7b6F3d9aEB473cE90fefA6170669616D
kubunt` block number:     1
kubunt` block timestamp: 1657824298
change` account:        0xBa10b9108e5Ff0b10A6439c61C680c04ccd058E0
` balance:          99.99915573025
11 pac` gas used:        250154 (0x3d12a)
run` gas price:        3.375 gwei
` value sent:        0 ETH
` total cost:        0.00084426975 ETH
found
kubunt` Saving migration to chain.
```

2:46 PM
7/14/22

```
- : sudo truffle -- Konsole
File Edit View Bookmarks Plugins Settings Help
## Run > Block gas limit: 6721975 (0x6691b7)
## You
## To 1_initial_migration.js
=====
e /usr
m/debi
----->
a8dada7cf76c
> transaction hash: 0x956de32e3d3fc2375c2045991ad537ce13d1e16aa703092b3dba
> Blocks: 0 Seconds: 0
> contract address: 0x91a2A73D7b6F3d9aEB473cE90fefA6170669616D
Readir
Buildi
Readir
nodejs
0 upgr
kubunt
kubunt
kubunt
nmp ->
v16.15
8.12.1
kubunt
kubunt
kubunt
Summary
change
11 pac
run
found
kubunt
truffle(develop)> undefined
truffle(develop)> create contract ProofOfExistence1
truffle(develop)>
```

6) Creating a Smart Contract

In the terminal, at the **truffle(develop)>** prompt execute :

create contract ProofOfExistence1

7) Adding Solidity Code

Open a New Terminal, execute :

```
cd
cd solidity-experiments
```

sudo nano contracts/ProofOfExistence1.sol

Remove the code you see in the above step and replace it with the code below and make you type this code don't copy paste if you run this code with error then you need to start from the beginning:

```
pragma solidity >= 0.5.0 < 0.7.0;
// Proof of Existence contract, version 1
contract ProofOfExistence1 {
    // state
    bytes32 public proof;
    // calculate and store the proof for a document
    // *transactional function*
    function notarize(string memory document) public {
        proof = proofFor(document);
    }
    // helper function to get a document's sha256
    // *read-only function*
    function proofFor(string memory document) public view returns (bytes32)
    {
        return sha256(bytes(document));
    }
}
```

8) Deploying the Contract

In the terminal execute the following command:

```
sudo nano migrations/2_deploy_contracts.js
```

Paste the below code in the file :

```
var ProofOfExistence1 = artifacts.require("./ProofOfExistence1.sol");
module.exports = function(deployer) {
  deployer.deploy(ProofOfExistence1);
};
```

The screenshot shows a Linux desktop environment with a terminal window open in the Konsole application. The terminal window has a title bar 'Proj 15 Ethereum with Truffle_2022'. Inside the terminal, a nano editor session is active for the file 'migrations/2_deploy_contracts.js'. The code shown in the editor matches the code provided in the question. The desktop taskbar at the bottom includes icons for a terminal, file manager, and browser.

Go to second terminal, with prompt ***truffle develop*** >, execute :

migrate -reset

The screenshot shows a terminal window titled "Proj 15 Ethereum with Truffle_2022" running on a Kubuntu desktop. The terminal output is as follows:

```
st/run.js:64:1
    ## Run
    ## You
    su
similar) it is a
## To i
only one sym
cu
e /usr/
ec
m/debia
su
With the new
back. So it is
years ago you
your data bas
0 upgra
kubuntu
SHA256 algor
example if yo
malware injec
website you a
Using this on
11 pack
run

found 0
kubuntu
Starting migrations...
=====
> Network name: 'develop'
> Network id: 5777
> Block gas limit: 6721975 (0x6691b7)

2_deploy_contracts.js
```

The terminal shows the command `migrate -reset` being run, followed by the compilation of the contract `ProofOfExistence1.sol`. It includes a warning about SPDX license identifiers and a note about function state mutability. The process then moves on to starting migrations, setting the network to 'develop', and defining the block gas limit.

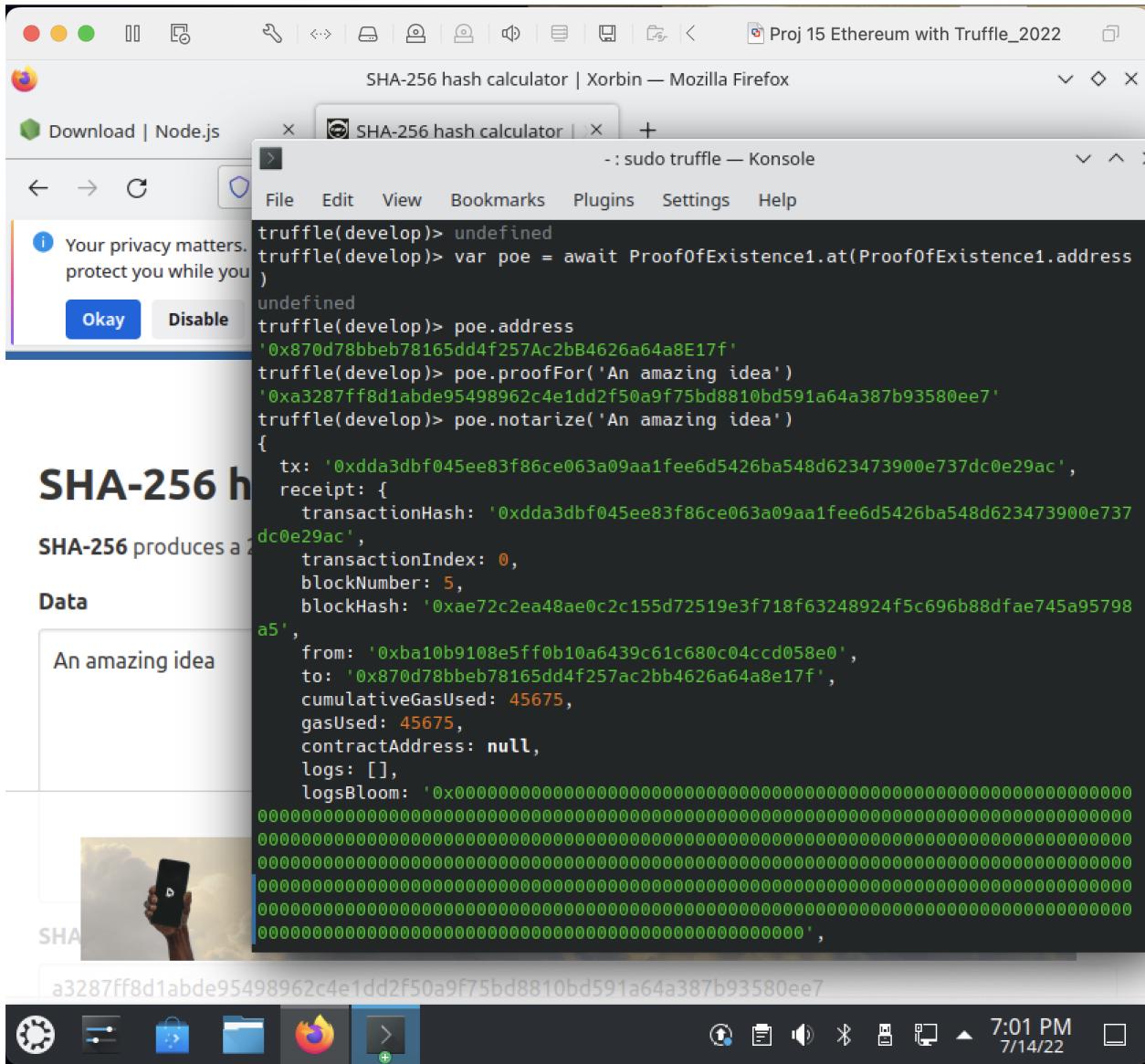
```
## Run
## You
Starting migrations...
=====
similar) it is al
## To i
    cu
    > Network name:      'develop'
only one sym
e /usr/
    ec
    > Network id:       5777
almost-unique
    ec
    > Block gas limit: 6721975 (0x6691b7)
m/debia
    su
2_deploy_contracts.js
=====
Reading
Buildin
Reading
nodejs
0 upgra
53159fde4c7a
SHA256 algor
example if yo
malware inject
website you a
kubuntu
SHA-256 is or
changed
Using this on
11 pack
run '
found 0
kubuntu
=====
> Saving migration to chain.
> Saving artifacts
=====
> Total cost:      0.000886926265918098 ETH
=====
Summary
=====
> Total deployments:   1
> Final cost:          0.000886926265918098 ETH
```

9) Finding the Smart Contract's Address

Same in the second terminal at **truffle develop** > prompt, execute :

```
var poe = await ProofOfExistence1.at(ProofOfExistence1.address)

poe.address
```



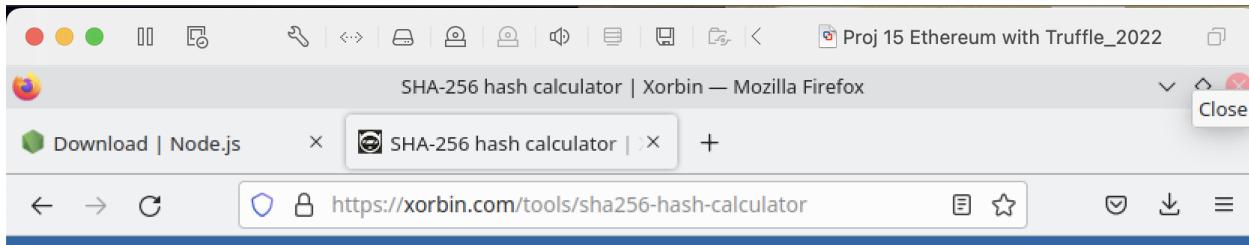
10) Calculating a Hash

Go to the web browser within your vm and go to this link :

<https://xorbin.com/tools/sha256-hash-calculator>

The SHA256 hash of this message is shown below.

In the original terminal, at the *truffle develop>* prompt, execute :



SHA-256 hash calculator

SHA-256 produces a 256-bit (32-byte) hash value.

Data

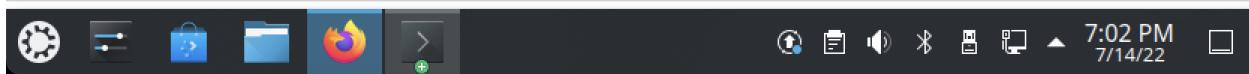
An amazing idea

SHA-256 hash

a3287ff8d1abde95498962c4e1dd2f50a9f75bd8810bd591a64a387b93580ee7

Hash added to your clipboard. Simply press ⌘+V, CTRL+V to paste.

Calculate SHA256 hash

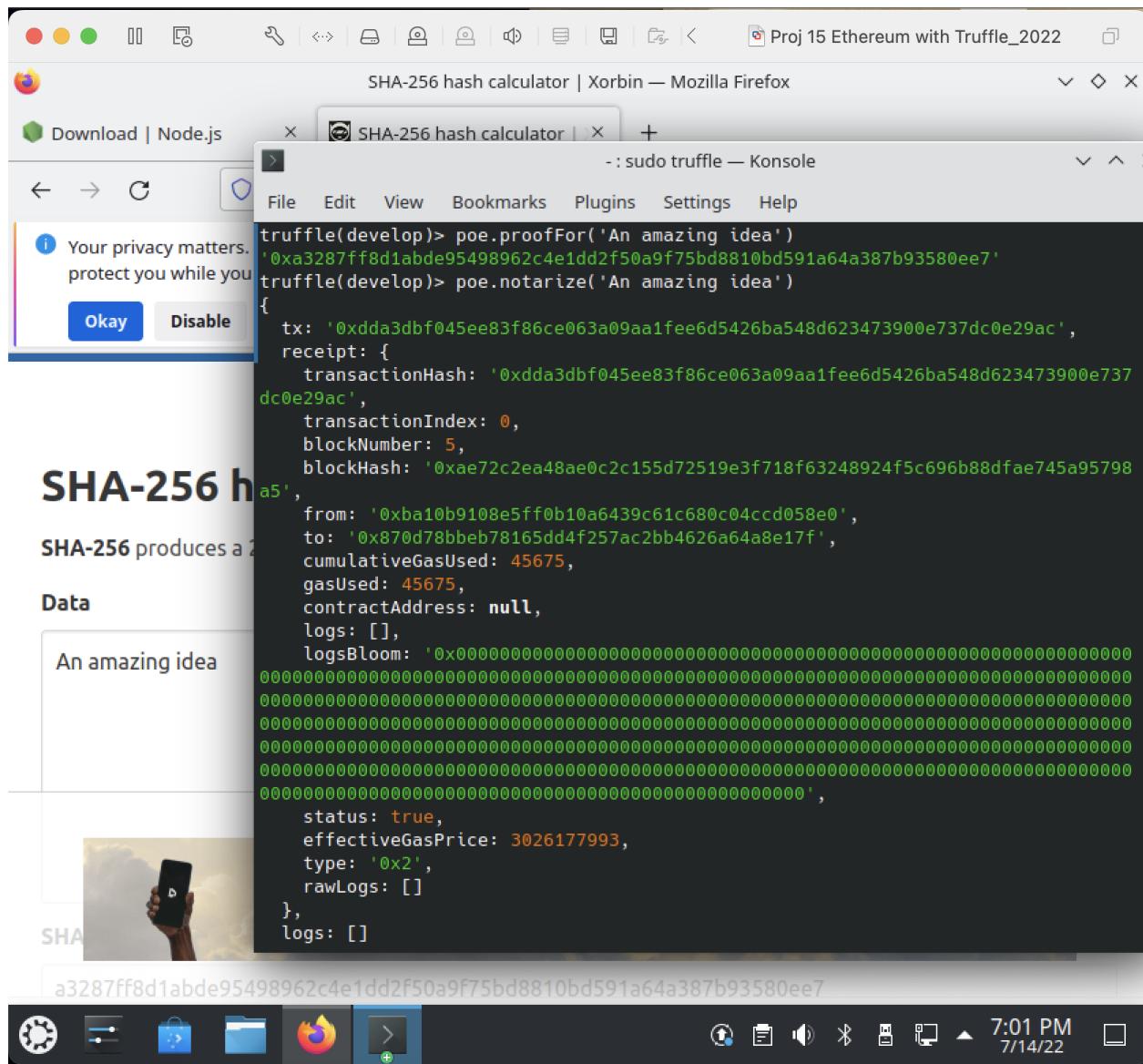


11) Registering a Document

In the original terminal, at the *truffle develop>* prompt, execute :

```
poe.notarize('An amazing idea')
```

```
poe.proof()
```



12) Registering Your Name

In the original terminal, at the ***truffle develop***> prompt, execute {replace YOUNAME with your name}:

```
poe.notarize('YOURNAME')
```

poe.proof()

CONCLUSION : We can conclude that first we install and set up truffle and installed dependencies then we initialize truffle and we develop it by truffle develop command then we compile it also we migrate it and we created Proofofexistence1 smart contract in which we add notarized transaction code. Then we reset it and deploy again. In the end we calculate a hash value of any string then we verify it on an online site and hashes completely match with it so we can say that we successfully made a smart contract in truffle.