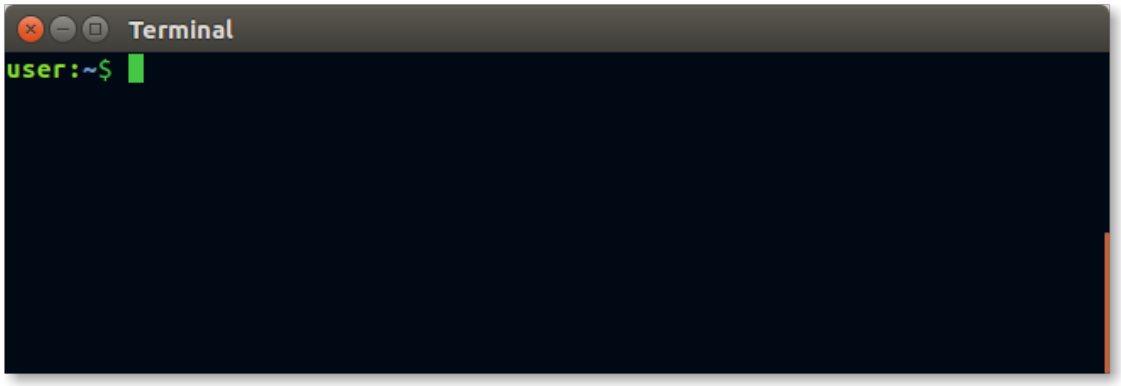
Setting up an Ethereum Playground

Choose your environment: Parity or Ethereum Wallet

Parity: We recommend using Parity for a smoother testing experience; the `--chain=dev` option simplifies the process considerably; it will automatically mine a new block whenever a contract call or ETH transfer is performed (which refrains from using your CPU any more than is needed), grant you test ETH to experiment with, and provide an all-around pleasant user-experience and user-interface for experimenting with different contract test deployments.

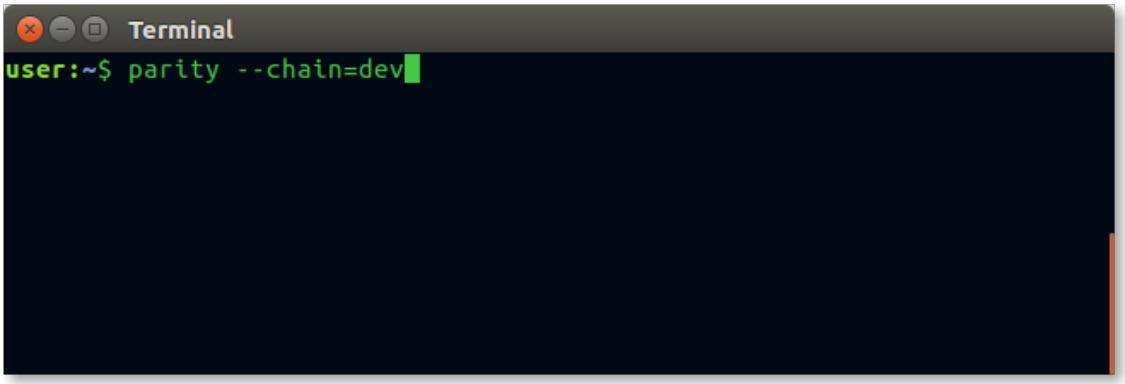
Download Parity: https://www.parity.io/

Open a terminal window. On Ubuntu, Ctrl+Alt+T is an easy shortcut to do this. You can also use the Launcher and type in “terminal” to find the application on your computer.



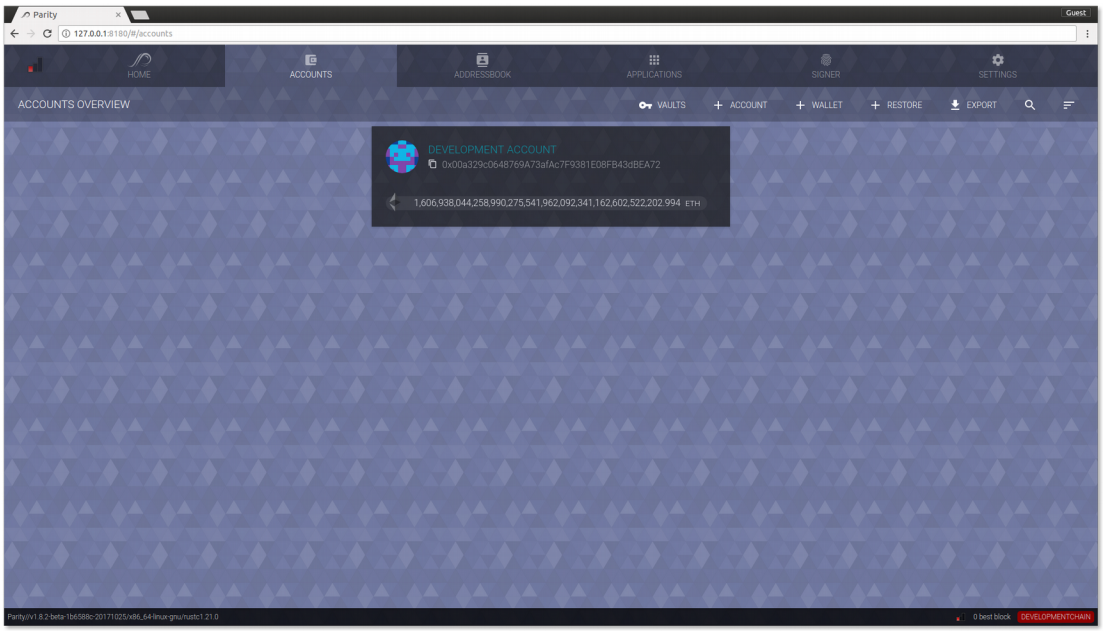
Run (in terminal):

```
parity --chain=dev
```

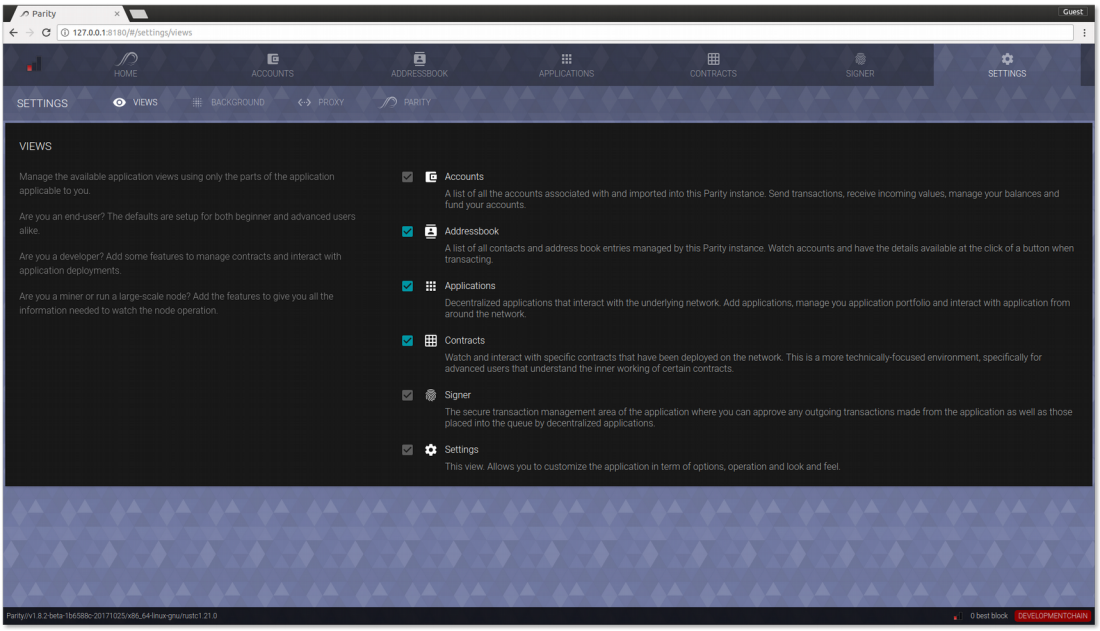


Open in your web browser: http://127.0.0.1:8180

Note: Once you've agreed to the Terms & Conditions, you can skip the account creation portion, because Parity automatically creates a Development Account pre-funded with a large ETH balance for testing. If you ever delete this account (or if it does not show up), you can Restore a new account from the Accounts page without entering any recovery phrase, and it will be granted the same allocation.



Use the "Settings" tab at the top to make sure the "Contracts" view tab is enabled.



Congratulations! You are now set up on a private testnet with Parity, ready to begin experimenting with Ethereum smart contracts.

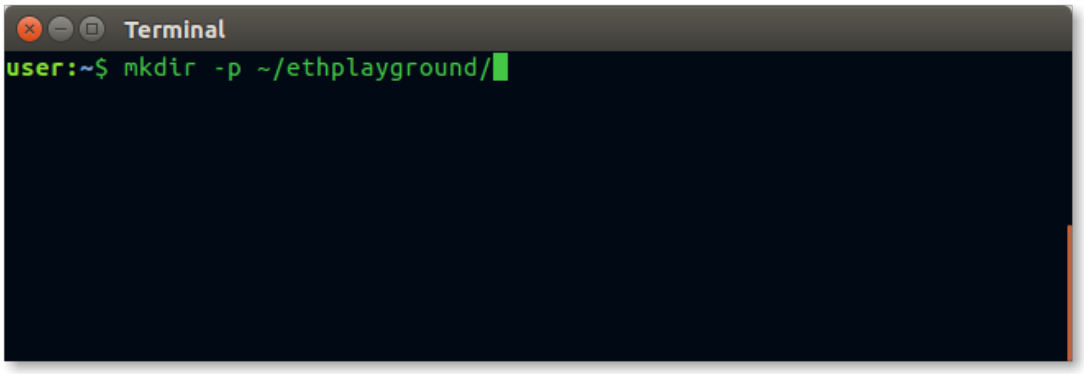
Ethereum Wallet (Mist) + geth: Using a geth-backend and Ethereum Wallet GUI is an alternative environment that can be used to test and experiment with contract deployments. It requires a little bit more effort to set up (the two applications must be linked together so that they can "talk to one another") and will require the user to either leave the miner running in the background (which can hog up the CPU) or to manually enable the mining function whenever any actions are performed that require a new block to be found in order to complete (like contract deployments, function calls, and ETH transfers).

Download geth: https://github.com/ethereum/go-ethereum/wiki/Building-Ethereum
Download Ethereum Wallet: https://github.com/ethereum/mist/releases

Open two separate terminal windows. On Ubuntu, Ctrl+Alt+T is an easy shortcut to open a terminal window. You can also use the Launcher and type in "terminal" to find the application on your computer. Either way, make sure to do it twice, so that two terminals are available.

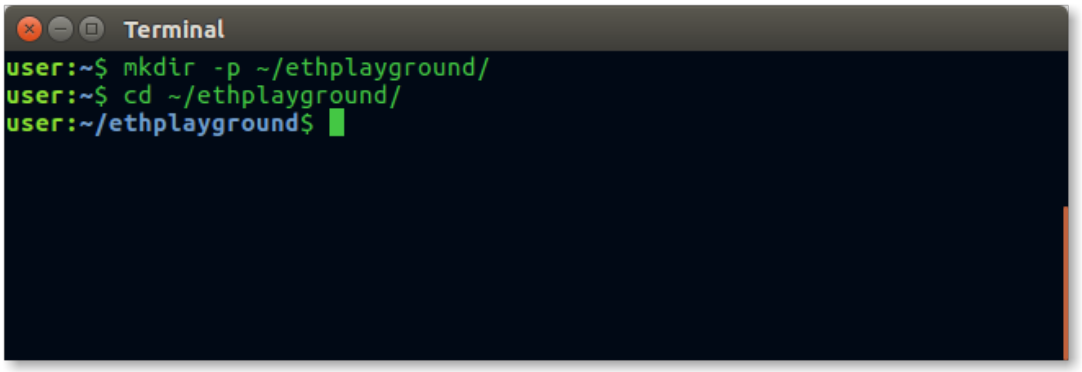
Set up data directory (in first terminal):

```
mkdir -p ~/ethplayground/
```



Run (in both terminals):

```
cd ~/ethplayground/
```

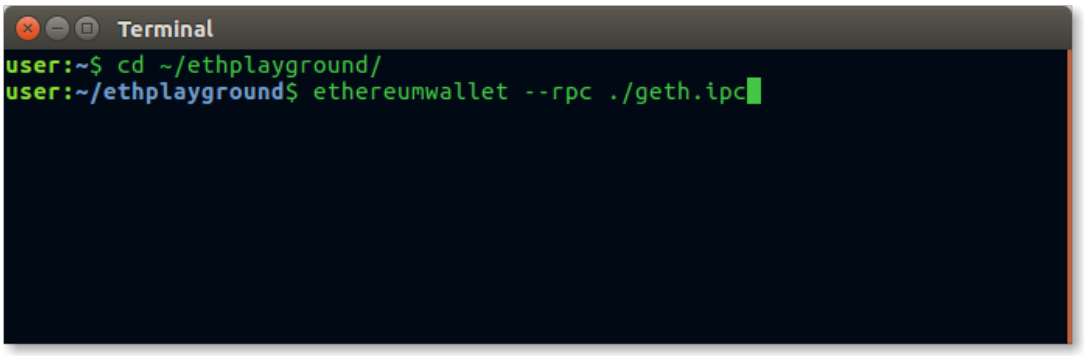


Run (in first terminal):

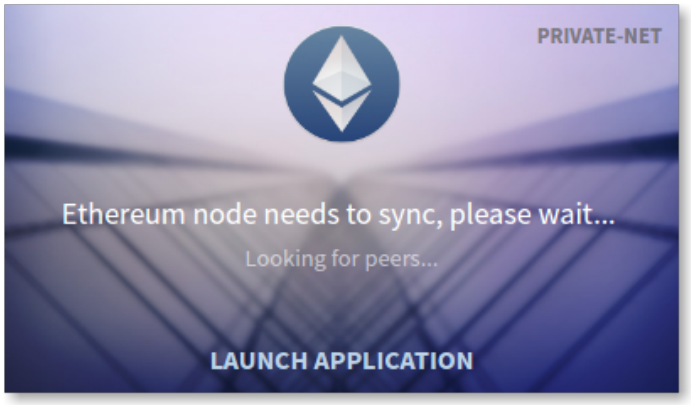
```
geth --dev --datadir ./ --ipcpath ./geth.ipc console
```

Run (in second terminal):

```
ethereumwallet --rpc ./geth.ipc
```

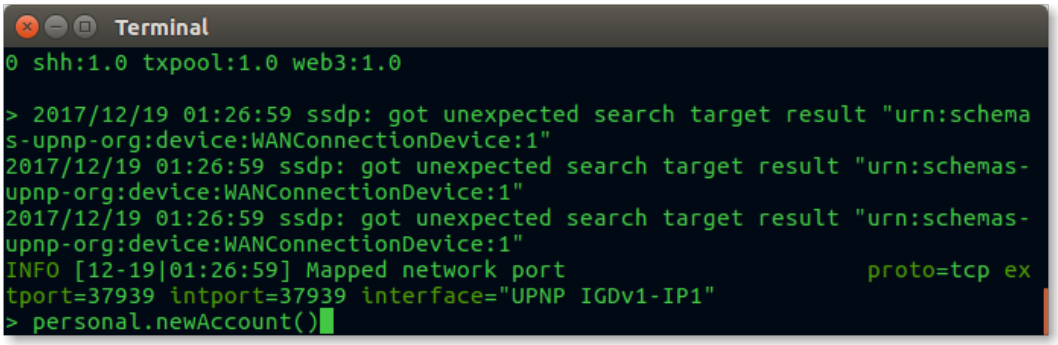


Note: an Ethereum Wallet application window should appear after running this command. After a moment, when it has loaded the configuration successfully, you will be able to click Launch Application to make the main window appear.



Run (in first terminal, on the geth console that is now open):

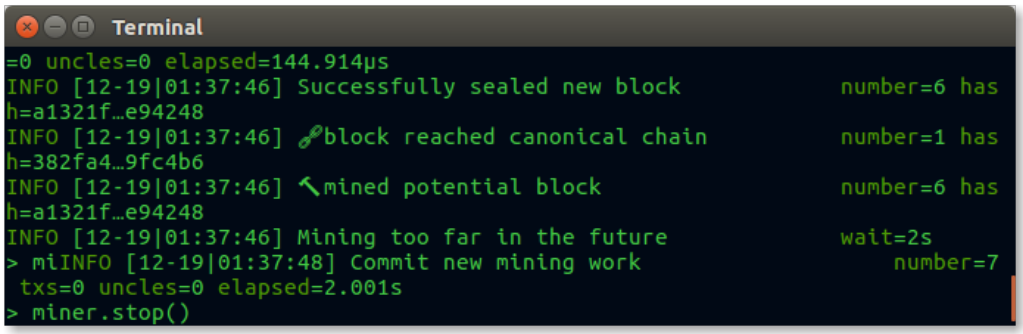
```
personal.newAccount()
```



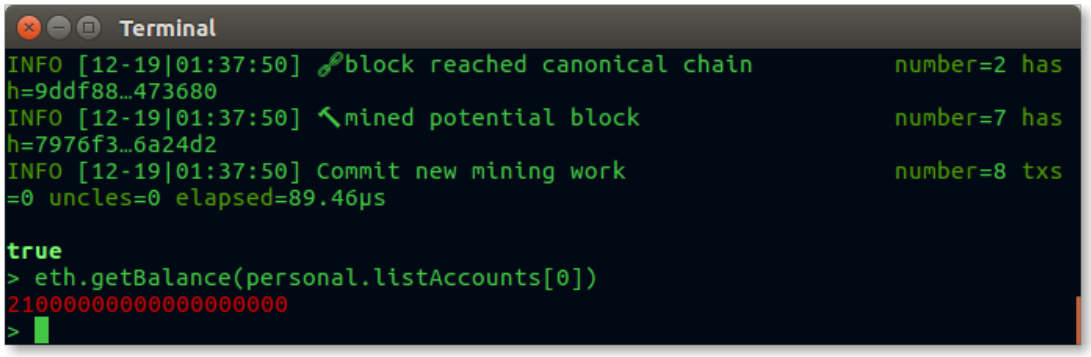

```
miner.start(1)
```

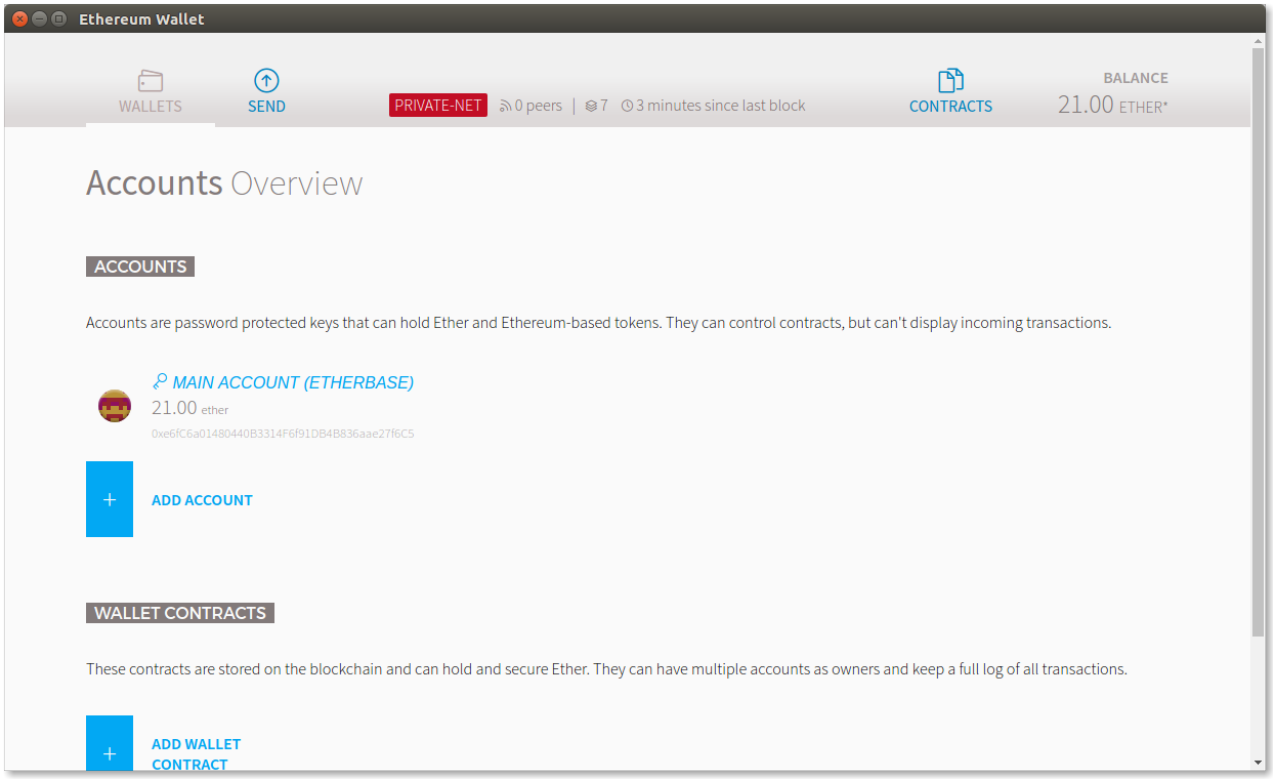
(wait a few seconds, so that a few blocks can be mined)

```
miner.stop()
```



Now you should have some ETH in your primary account address (you can verify this in the geth console with the command `eth.getBalance(personal.listAccounts[0])` or in the Ethereum Wallet application window, both of which should show a non-zero balance at this point).





Congratulations! You are now set up on a private geth testnet with Ethereum Wallet, and you are ready to begin experimenting with Ethereum smart contracts.