

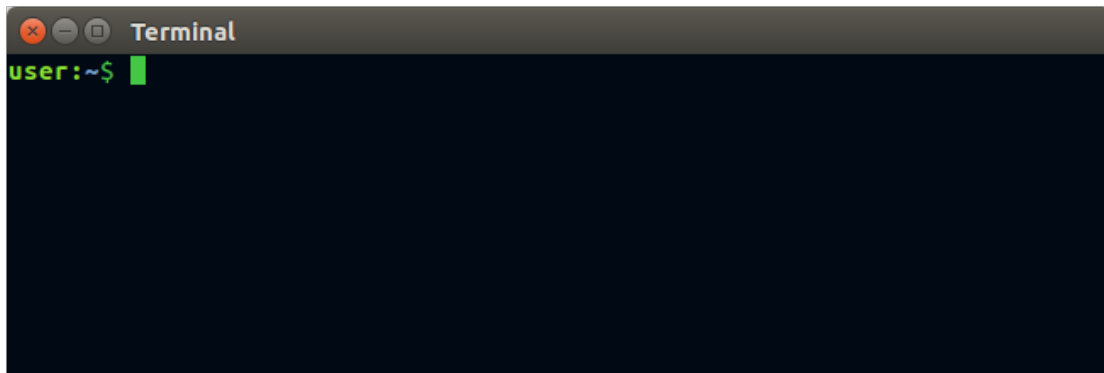
Setting up an Ethereum Playground

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.

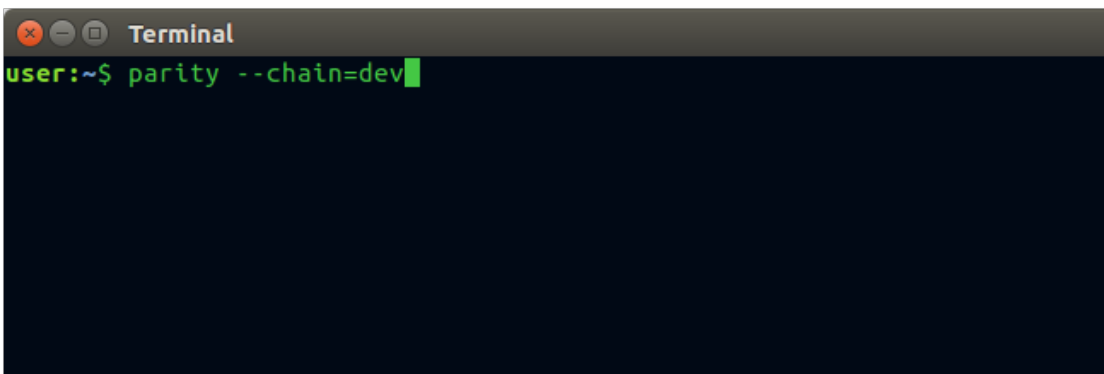
Download Parity: <https://www.parity.io/>

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.



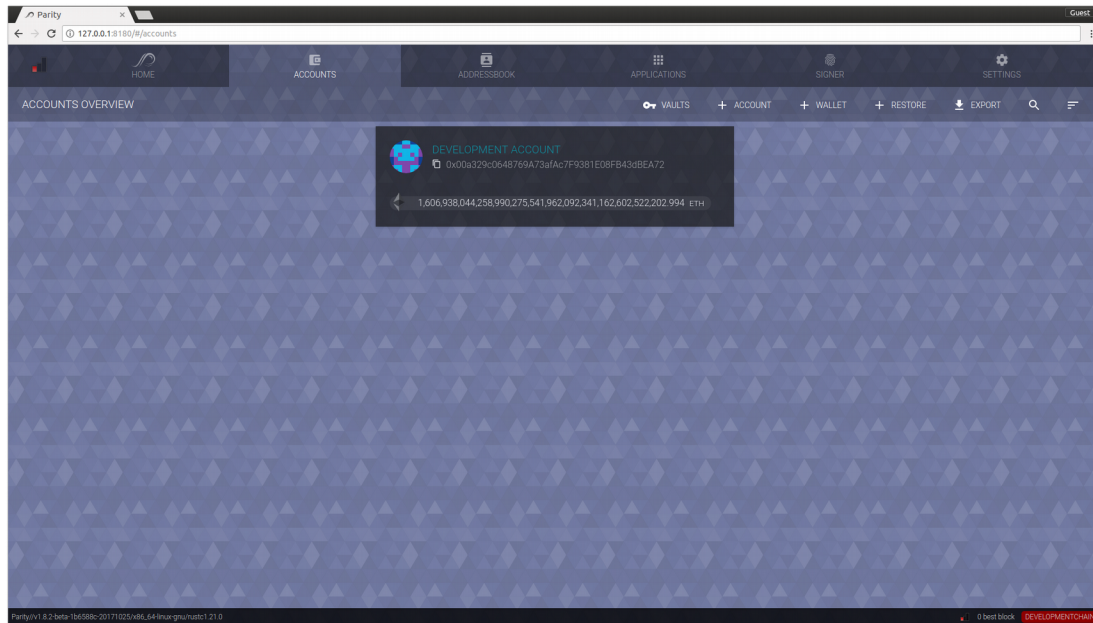
Run (in terminal):

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

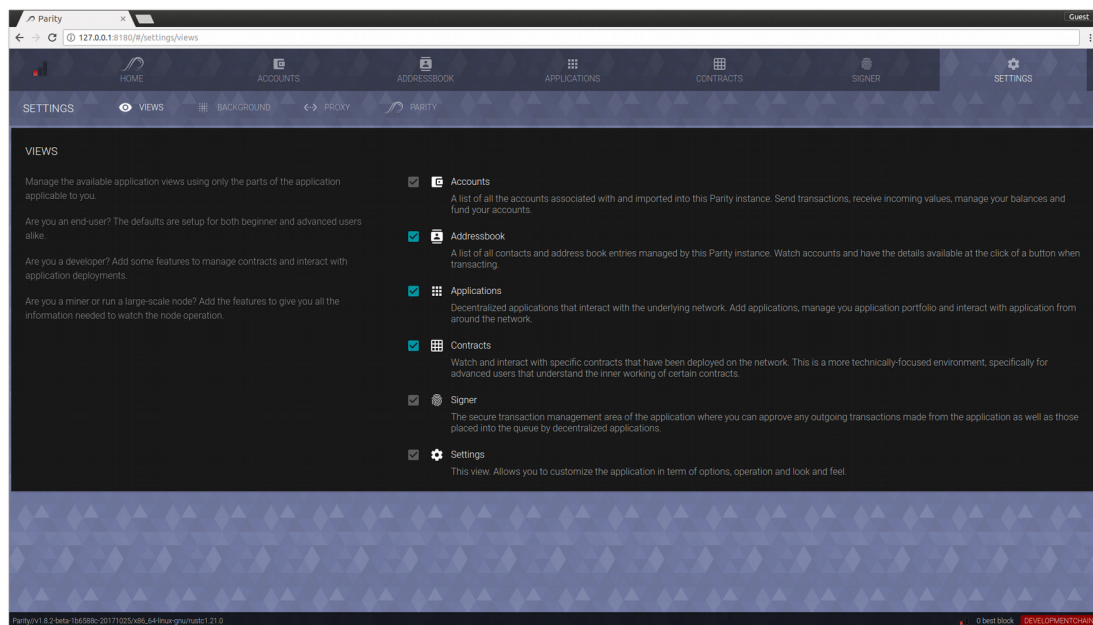


Open in your web browser: <http://127.0.0.1:8180>

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).

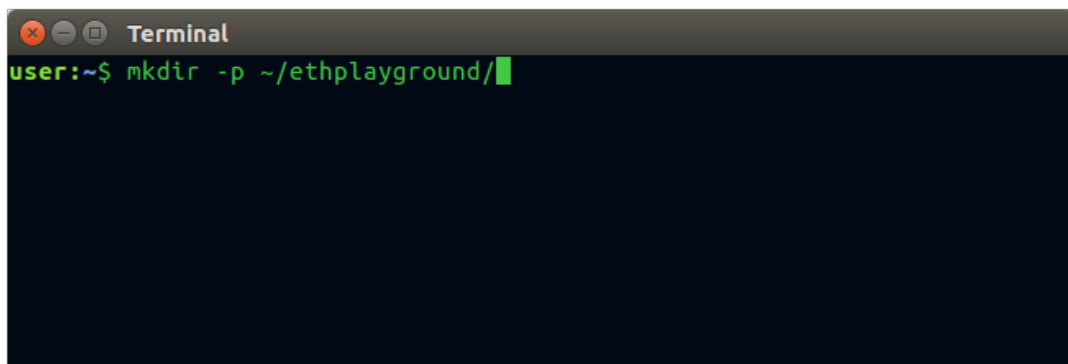
Download geth: <https://github.com/ethereum/go-ethereum/wiki/Building-Ethereum>

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.

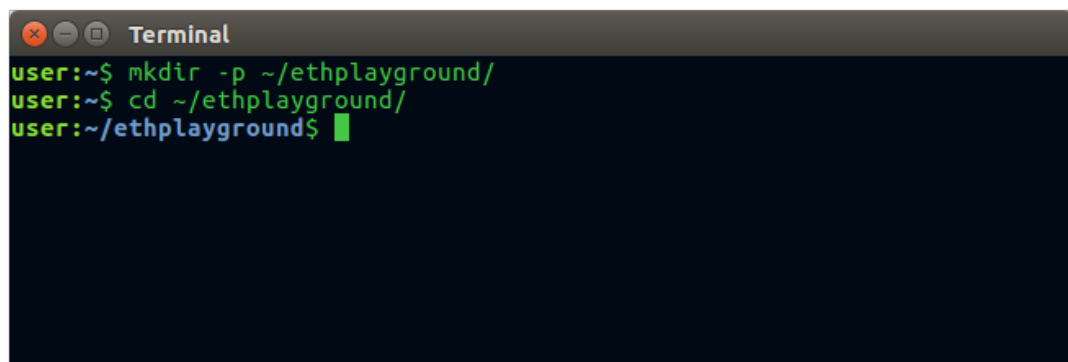
Set up data directory (in first terminal):

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

A screenshot of a terminal window with a dark background. The title bar says "Terminal". The prompt is "user:~\$". The command "mkdir -p ~/ethplayground/" has been entered and executed, with a green cursor at the end of the line.

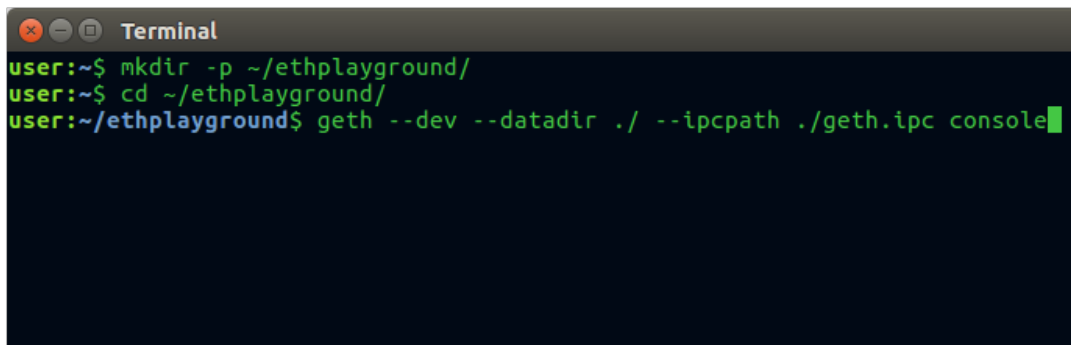
Run (in both terminals):

```
cd ~/ethplayground/
```

A screenshot of a terminal window with a dark background. The title bar says "Terminal". The prompt is "user:~\$". The command "mkdir -p ~/ethplayground/" has been entered and executed. The prompt is now "user:~\$". The command "cd ~/ethplayground/" has been entered and executed. The prompt is now "user:~/ethplayground\$".

Run (in first terminal):

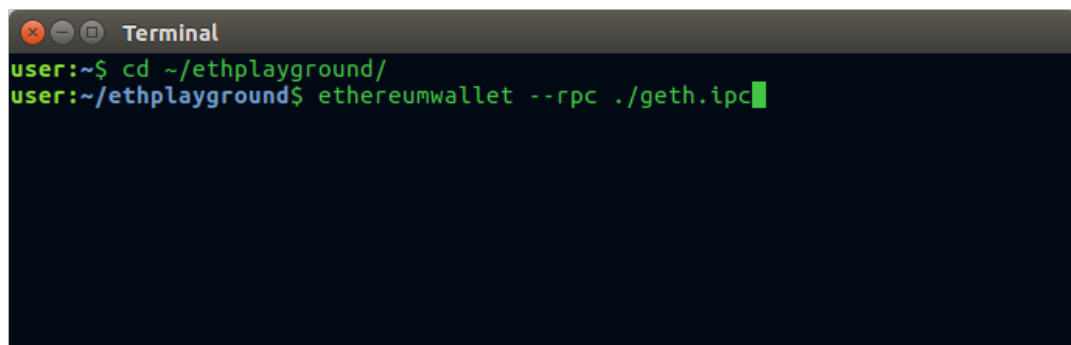
```
geth --dev --datadir ./ --ipcpath ./geth.ipc console
```



```
Terminal
user:~$ mkdir -p ~/ethplayground/
user:~$ cd ~/ethplayground/
user:~/ethplayground$ geth --dev --datadir ./ --ipcpath ./geth.ipc console
```

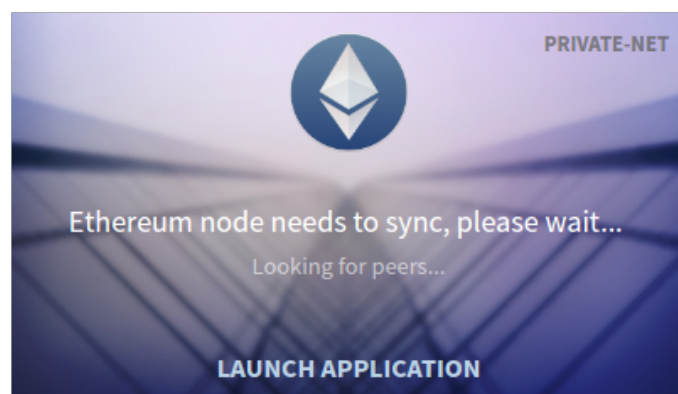
Run (in second terminal):

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



```
Terminal
user:~$ cd ~/ethplayground/
user:~/ethplayground$ 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()
```

A terminal window titled "Terminal" showing network logs and a command. The logs include timestamps and messages about SSDP search results and network port mapping. The command `personal.newAccount()` is entered at the prompt.

```
0 ssh:1.0 txpool:1.0 web3:1.0
> 2017/12/19 01:26:59 ssdp: got unexpected search target result "urn:schema
s-upnp-org:device:WANConnectionDevice:1"
2017/12/19 01:26:59 ssdp: got unexpected search target result "urn:schemas-
upnp-org:device:WANConnectionDevice:1"
2017/12/19 01:26:59 ssdp: got unexpected search target result "urn:schemas-
upnp-org:device:WANConnectionDevice:1"
INFO [12-19|01:26:59] Mapped network port                proto=tcp ex
tport=37939 intport=37939 interface="UPNP IGDv1-IP1"
> personal.newAccount()
```

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

A terminal window titled "Terminal" showing network logs, the command `personal.newAccount()`, and the command `miner.start(1)`. The logs include timestamps and messages about SSDP search results and network port mapping. The command `personal.newAccount()` prompts for a passphrase, and `miner.start(1)` is entered at the prompt.

```
upnp-org:device:WANConnectionDevice:1"
2017/12/19 01:26:59 ssdp: got unexpected search target result "urn:schemas-
upnp-org:device:WANConnectionDevice:1"
INFO [12-19|01:26:59] Mapped network port                proto=tcp ex
tport=37939 intport=37939 interface="UPNP IGDv1-IP1"
> personal.newAccount()
Passphrase:
Repeat passphrase:
"0xe6fc6a01480440b3314f6f91db4b836aae27f6c5"
> miner.start(1)
```

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

```
miner.stop()
```

A terminal window titled "Terminal" showing mining logs and the command `miner.stop()`. The logs include timestamps and messages about block sealing, canonical chain, mining potential, and mining work. The command `miner.stop()` is entered at the prompt.

```
=0 uncles=0 elapsed=144.914µs
INFO [12-19|01:37:46] Successfully sealed new block          number=6 has
h=a1321f...e94248
INFO [12-19|01:37:46] block reached canonical chain          number=1 has
h=382fa4...9fc4b6
INFO [12-19|01:37:46] mined potential block                  number=6 has
h=a1321f...e94248
INFO [12-19|01:37:46] Mining too far in the future           wait=2s
> miINFO [12-19|01:37:48] Commit new mining work                 number=7
txs=0 uncles=0 elapsed=2.001s
> miner.stop()
```

[illegible]