

Run SUAVE Locally

You can use `suave-geth` to start a local SUAVE devnet.

There are two ways to work with `suave-geth` :

1. > [Install the latest release binary](#)
2. ♀ [Build from source](#)

Latest release binary

`curl`

`-L https://suaveup.flashbots.net |`

`bash` Start your local devnet with:

`suave-geth --suave.dev`

Building from source

info We recommend that you use Golang v1.21 or newer. Clone the [suave-geth](#) repository:

`git clone git@github.com:flashbots/suave-geth.git` `cd suave-geth` Build the binary:

`make suave` Now you have a `suave` binary in `./build/bin/` :

`./build/bin/suave-geth --version` Start the local devnet with:

`./build/bin/suave-geth --suave.dev` Missing packages If you have set up a new machine to run through this, you'll also need to install `Make` and `Go`: `sudo apt install make golang-go`

Testing the devnet

Compile the example contracts:

What is Forge? `forge` is a part of the smart contract development toolchain we use in our examples, which you can learn more about in our [next tutorial](#) . If you do not have it installed, you can do so quickly with: `curl -L https://foundry.paradigm.xyz | bash` `cd suave` && `forge build` Create a few example transactions:

`go run devenv/cmd/main.go` Execute a RPC request with `curl` like this:

`curl`

`'http://localhost:8545'`

`--header`

`'Content-Type: application/json'`

`--data`

`{ "jsonrpc": "2.0", "method": "eth_blockNumber", "params": [], "id": 83 }`

If you built from source (but not if you're running Docker), you can attach to the usual Geth javascript console to get any interactive data you need with:

`./build/bin/suave-geth attach /tmp/geth.ipc` IPC not found It may be the case, especially if you have a custom `GOPATH`, that the `geth.ipc` is somewhere else. You can either force the path using the `--ipcpath` flag when starting SUAVE, or find the path in the logs (it could look something like `/System/Volumes/Data/private/var/folders/rt/gq5bhvy17wz5z5dl32_x83dw0000gp/T/geth.ipc` , for instance). From within the console, you can run:

`eth.accounts [0]` This should return `"0xb5feafbdd752ad52afb7e1bd2e40432a485bbb7f"` - the default funded account for local development.

`eth.getBalance (eth.accounts [0])` Should return a really large number `1.1579...e+77` . If you try `eth.getBalance("")` instead, you should see `0` .

If you try:

web3.eth.blockNumber It should tell you the block height of your local network.

What am I actually running?

The main actor in the SUAVE protocol is called a "Kettle". Kettles house all components necessary to perform confidential compute.

Here is the architecture of a Kettle on the Rigil Testnet. When you start a local SUAVE devnet, you're running all the stuff in the purple square (but not the domain specific services, i.e. nodes connected to other chains from/to which you wish to receive or send bundles).

You can read more about exactly what a Kettle contains in [architecture section of the Technical Specs](#). [Edit this page](#)
[Previous Tutorials](#) [Next Deploy Contracts](#)