Using SUAVE Testnets

Deploy Contracts

Thespell tool we've been using in all our previous tutorials is only intended for local use. Therefore, we'll need to adapt our approach slightly in order to get our contract working on the testnet, and start transacting there.

1. Get rETH from thefaucet
2. .
3. Use Forge. From the root of your contracts directory, you can run:

forge create --rpc-url https://rpc.toliman.suave.flashbots.net --legacy

\ --private-key < your_funded_priv_key

< your_contract_name

.sol: < your_contract_name

info Note the--legacy flag: transactions on SUAVE are not EIP1559 compliant, which you will likely need to take into account no matter which smart contract framework you use. You should see something like this printed to your console:

.. .relevant compilation results .. . Deployer: 0xBE69d72ca5f88aCba033a063dF5DBe43a4148De0 Deployed to: 0xcbdF0322Cd79212e10b0dB72D775aE05B99c1796 Transaction hash: 0x9ae80af40bdafbc706108446dbbf7761a59f5bf544b46c97b9b0851dddaa3927

Sending Transactions

We support both aGolang SDK and atypescript SDK to help you deploy contracts and send confidential compute requests.

Typescript SDK

In theprevious tutorial , we demonstrated how to use this SDK to craft CCRs in your local environment. The process is exactly the same for our SUAVE testnet: you just need to change the proivder url (and you can use the developer Kettle for sending transactions if the faucet doesn't work):

const

SUAVE_RPC_URL

=

'http://localhost:8545' ; ... const

PRIVATE_KEY :

Hex

= '0x6c45335a22461ccdb978b78ab61b238bad2fae4544fb55c14eb096c875ccfc52' ; Otherwise the process is exactly the same as in your local environment.

Golang SDK

The most effective way to begin with the Golang SDK is to use the sameframework.go used for all thesuapp-examples , as it implements everything you need for interacting with your contracts.

Its use is demonstrated in themain.go in each example, from which you should be able to learn everything from deploying contracts to sending custom confidential compute requests from various different actors. Edit this page Previous Turn Existing Contracts Into Suapps Next Cross Chain Suapps