
title: Networks description: An overview of Ethereum's networks and where to get testnet ether (ETH) for testing your application. lang: en

Ethereum networks are groups of connected computers that communicate using the Ethereum protocol. There is only one Ethereum Mainnet, but independent networks conforming to the same protocol rules can be created for testing and development purposes. There are many independent "networks" that conform to the protocol without interacting with each other. You can even start one locally on your own computer for testing your smart contracts and web3 apps.

Your Ethereum account will work across the different networks, but your account balance and transaction history won't carry over from the main Ethereum network. For testing purposes, it's useful to know which networks are available and how to get testnet ETH to play around with. In general, for security considerations, it's not recommended to reuse mainnet accounts on testnets or vice versa.

Prerequisites {#prerequisites}

You should understand the [basics of Ethereum](#) before reading up on the different networks, as the test networks will give you a cheap, safe version of Ethereum to play around with.

Public networks {#public-networks}

Public networks are accessible to anyone in the world with an internet connection. Anyone can read or create transactions on a public blockchain and validate the transactions being executed. The consensus among peers decides on the inclusion of transactions and the state of the network.

Ethereum Mainnet {#ethereum-mainnet}

Mainnet is the primary public Ethereum production blockchain, where actual-value transactions occur on the distributed ledger.

When people and exchanges discuss ETH prices, they're talking about Mainnet ETH.

Ethereum Testnets {#ethereum-testnets}

In addition to Mainnet, there are public testnets. These are networks used by protocol developers or smart contract developers to test both protocol upgrades as well as potential smart contracts in a production-like environment before deployment to Mainnet. Think of this as an analog to production versus staging servers.

You should test any contract code you write on a testnet before deploying to Mainnet. Among dapps that integrate with existing smart contracts, most projects have copies deployed to testnets.

Most testnets started by using a permissioned proof-of-authority consensus mechanism. This means a small number of nodes are chosen to validate transactions and create new blocks – staking their identity in the process. Alternatively, some testnets feature an open proof-of-stake consensus mechanism where everyone can test running a validator, just like Ethereum Mainnet.

ETH on testnets is supposed to have no real value; however, there have been markets created for certain types of testnet ETH that have become scarce or hard to obtain. Since you need ETH to actually interact with Ethereum (even on testnets), most people get testnet ETH for free from faucets. Most faucets are webapps where you can input an address which you request ETH to be sent to.

Which Testnet should I use?

The two public testnets that client developers are currently maintaining are Sepolia and Goerli. Sepolia is a network for contract and application developers to test their applications. The Goerli network lets protocol developers test network upgrades, and lets stakers test running validators.

Sepolia {#sepolia}

Sepolia is the recommended default testnet for application development The Sepolia network uses a permissioned validator set. It's fairly new, meaning its state and history are both quite small. This means the network is quick to sync to and that running a node on it requires less storage. This is useful for users who want to quickly spin up a node and interact with the network directly.

- Closed validator set, controlled by client & testing teams
- New testnet, less applications deployed than other testnets
- Fast to sync and running a node requires minimal disk space

Resources

- [Website](#)
- [GitHub](#)
- [Otterscan](#)
- [Etherscan](#)

Faucets

- [QuickNode Sepolia Faucet](#)
- [Grabteeth](#)
- [PoW faucet](#)
- [Coinbase Wallet Faucet | Sepolia](#)
- [Alchemy Sepolia faucet](#)
- [Infura Sepolia faucet](#)
- [Chainstack Sepolia faucet](#)
- [Testnet Faucet | Sepolia](#)

Goerli (*long-term support*) {#goerli}

Note: [the Goerli testnet is deprecated](#) and will be replaced by [Holesovice](#) in 2023. Please consider migrating your applications to Sepolia.

Goerli is a testnet for testing validating and staking. The Goerli network is open for users wanting to run a testnet validator. Stakers wanting to test protocol upgrades before they are deployed to mainnet should therefore use Goerli.

- Open validator set, stakers can test network upgrades
- Large state, useful for testing complex smart contract interactions
- Longer to sync and requires more storage to run a node

Resources

- [Website](#)
- [GitHub](#)
- [Etherscan](#)

Faucets

- [QuickNode Goerli Faucet](#)
- [Grabteeth](#)
- [PoW faucet](#)
- [Paradigm faucet](#)
- [Alchemy Goerli Faucet](#)
- [All That Node Goerli Faucet](#)
- [Coinbase Wallet Faucet | Goerli](#)
- [Chainstack Goerli faucet](#)

To launch a Validator on Goerli testnet, use ethstaker's "[cheap goerli validator](#)" [launchpad](#).

Layer 2 testnets {#layer-2-testnets}

[Layer 2 \(L2\)](#) is a collective term to describe a specific set of Ethereum scaling solutions. A layer 2 is a separate blockchain that extends Ethereum and inherits the security guarantees of Ethereum. Layer 2 testnets are usually tightly coupled to public Ethereum testnets.

Arbitrum Goerli {#arbitrum-goerli}

A testnet for [Arbitrum](#).

Faucets

- [Chainlink faucet](#)

Optimistic Goerli {#optimistic-goerli}

A testnet for [Optimism](#).

Faucets

- [Paradigm faucet](#)
- [Coinbase Wallet Faucet | Optimism Goerli](#)

Starknet Goerli {#starknet-goerli}

A testnet for [Starknet](#).

Faucets

- [Starknet faucet](#)

Private networks {#private-networks}

An Ethereum network is a private network if its nodes are not connected to a public network (i.e. Mainnet or a testnet). In this context, private only means reserved or isolated, rather than protected or secure.

Development networks {#development-networks}

To develop an Ethereum application, you'll want to run it on a private network to see how it works before deploying it. Similar to how you create a local server on your computer for web development, you can create a local blockchain instance to test your dapp. This allows for much faster iteration than a public testnet.

There are projects and tools dedicated to assist with this. Learn more about [development networks](#).

Consortium networks {#consortium-networks}

The consensus process is controlled by a pre-defined set of nodes that are trusted. For example, a private network of known academic institutions that each govern a single node, and blocks are validated by a threshold of signatories within the network.

If a public Ethereum network is like the public internet, a consortium network is like a private intranet.

Related tools {#related-tools}

- [Chainlist](#) *list of EVM networks to connect wallets and providers to the appropriate Chain ID and Network ID*
- [EVM-based Chains](#) *GitHub repo of chain metadata that powers Chainlist*

Further reading {#further-reading}

- [Proposal: Predictable Ethereum Testnet Lifecycle](#)
- [The Evolution of Ethereum Testnets](#)