Upgrade GoQuroum

The following information is for upgrading GoQuorum only.

You can have multiple versions of GoQuorum running on the same network, allowing you to perform a rolling (node-by-node) upgrade. Ensure your network maintains full uptime availability when upgrading to provide redundancy and reversibility if anything goes wrong.

Incremental upgrades

Stop the GoQuorum node you wish to upgrade, update the binary to the next version, and restart the node. Check that the node starts without errors and has the latest block from the network.

We don't recommend jumping versions during an upgrade; some versions require manual intervention. Check the release log for any actions you might need to take.

Upgrade to GoQuorum 22.7.1

This release is an upstream merge of Geth v1.10.3 there are a couple of changes to note.

Support for eth/64 has been removed

The minimum protocol version is now eth/65 If you wish to upgrade any client in your network to this version all other clientsshould be running at least v21.4.0 . Clients running prior to this version will continue to function if there are other nodes running on the network that support eth/64 and eth/65 (for example, v21.4.0 through v22.7.1). However, they will stop receiving block headers and transactions if this is not the case. For example if nodeA and nodeB are running v22.7.1 while nodeC is running v21.1 - nodeC will not be able to communicate with nodeA and nodeB.

Berlin Hard Fork

It is no longer possible to upgrade snapshot databases generated by pre-Berlin geth #22663 . You must specify a block forberlinBlock if you wish to use snap sync.

caution If you are using any privacy features you must use--syncmode full

Upgrade to GoQuorum 22.4.1

When upgrading to and past this version, you can specify a block forberlinBlock as the milestone block in your genesis file and rungeth init again. The block needs to be sufficiently in the future to allowALL your nodes in the network to upgrade before that block height is reached.

If you do not specify a value forberlinBlock , it is not applied. However, specifying aberlinBlock is required for some features (such as snap sync) to continue working and for subsequent forks to be applied (for example, london fork).

For more details about EIPs included in this upgrade and how they impact gas calculations, refer to the Ethereum blog .

Upgrade to GoQuorum 22.1.0

There are several significant changes to the underlying Geth 1.10 in the release of GoQuorum 22.1.0. Before you upgrade to this GoQuorum version, consider and prepare for these changes. The main things to be aware of are option deprecations and chain ID enforcement .

Option deprecations

Several command line interface (CLI) options are deprecated in the 1.9.x releases of Geth (first merged into GoQuorum as of version 2.6.0). Support for deprecated options is removed in GoQuorum 22.1.0.

See the full list of deprecated options .

Chain ID enforcement

Signing transactions with a chain ID is enforced as the default behavior in Geth 1.10 and GoQuorum 22.1.0 for raw transactions. This impacts client libraries, tooling, and applications.

You can send signed raw private transactions that include a chain ID to GoQuorum, but when you use sendTransaction for private transactions, GoQuorum continues to sign it using the Homestead signer (that is, without chain ID).

Recommended changes

We recommend you sign transactions using the chain ID on your network. If you use a client library, for example,web3j-quorum , you must use the latestQuorumTransactionManager to specify a chain ID. Withweb3js-quorum you can add thechainId property to the transaction, as seen in theQuorum Developer Quickstart examples.

If you don't want to make application changes

From version 22.1.0, you can start GoQuorum with the option--rpc.allow-unprotected-txs . This allows you to submit transactions that aren't signed with a chain ID.

Upgrade to GoQuorum 2.6.0

GoQuorum 2.6.0 upgrades the base Geth version from 1.8.18 to 1.9.7. SeeEthereum 1.9.0 for the complete list if new features added as a part of Geth 1.9.7.

note Geth 1.9.7 has several enhancements at the database layer which are part of GoQuorum 2.6.0. Once you migrate to 2.6.0, it can't roll back to an older version of GoQuorum. We recommend backing up the data directory before upgrading to 2.6.0, which you can use to revert back to the older version if necessary. A node running on GoQuorum 2.6.0 can coexist in a network with other nodes on lower versions of GoQuorum, so you can perform a rolling upgrade to GoQuorum 2.6.0. The following is the recommended upgrade process:

1. Stop the node you wish to upgrade to GoQuorum 2.6.0. Modify thegenesis.json
2. file to includeistanbulBlock
3. andpetersburgBlock
4. . Set these parameters to an appropriate value in future by when the entire network will be upgraded to GoQuorum 2.6.0.
5. caution
6. Setting the milestone blocks is necessary because the gas calculation logic changes in Geth 1.9.7. Not setting these values properly can result in aBad block error
7. .
8. GoQuorum 2.6.0 deprecates genesis parametersmaxCodeSize
9. andmaxCodeSizeChangeBlock
10. . A new attributemaxCodeSizeConfig
11. is added to genesis to allow tracking of multiplemaxCodeSize
12. value changes.
13. In earlier GoQuorum versions, if themaxCodeSize
14. is changed multiple times, any node joining the network might get aBad block error
15. . The changes in GoQuorum 2.6.0 enable tracking of historical changes ofmaxCodeSize
16. in the genesis file and thus allow it to be changed multiple times in the network's life. Wheninit
17. is executed in GoQuorum 2.6.0, Geth forces use ofmaxCodeSizeConfig
18. .
19. "config"
20. :
21. {
22. "homesteadBlock"
23. :
24. 0
25. ,
26. "byzantiumBlock"
27. :
28. 0
29. ,
30. "constantinopleBlock"
31. :
32. 0
33. ,
34. "petersburgBlock"
35. :
36. 0
37. ,
38. "istanbulBlock"
39. :
40. 0
41. ,
42. "chainId"
43. :
44. 10

,
"eip150Block"
:
0
,
"eip155Block"
:
0
,
"eip150Hash"
:
"0x0000000000000000000000000000000000000000000000000000000000000000"
,
"eip158Block"
:
0
,
"isQuorum"
:
true
,
"maxCodeSizeConfig"
:
[
{
"block"
:
5
,
"size"
:
35
}
,
{
"block"
:
15
,
"size"
:
24
}
,
{
"block"
:
20
,
"size"
:
35
}
]
}
,
Executegeth --datadir path/to/datadir init genesis.json
with the modifiedgenesis.json
.
Start the node in GoQuorum 2.6.0.

Notes * Geth 1.9.7 introducesfreezerdb * , where block data beyond a certain threshold is moved to a different file-based storage area. Provide the location forfreezerdb * using the--datadir.ancient * Geth command line option. * When a node is migrated to this version, Geth by default creates theancient * data folder and starts moving blocks below the immutability threshold (default: 3162240) into the ancient data. * If you don't want this movement to happen, use--immutabilitythreshold * to set the immutability threshold to an appropriate value when starting Geth. * Geth 1.9.7 by default doesn't allow unlocking keystore-based accounts in the startup process, and crashes if you attempt this. To enable account unlocking, use the--allow-insecure-unlock * Geth option. * In a GoQuorum 2.6.0 network running withgcmode=full * and block height exceeding the immutability threshold (with blocks infreezerdb *), if a node is restarted non-gracefully (kill -9/docker kill & start *), it can

fail to sync up with its peers, generating amissing parent * error. This is due to an upstream bug where non-graceful restart causes a gap betweenleveldb * andfreezerdb * . * You can avoid this by either running the node withgcmode=archive * or restarting the node gracefully (kill / docker stop & start *). * This is fixed in GoQuorum v21.4.0 (from upstream Geth 1.9.20).
Edit this page Last updatedonMay 15, 2023 byAlexandra Tran Carrillo Previous Kubernetes Next Deployment examples