Handling DKG failure

While the DKG process has been tested and validated against many different configuration instances, it can still encounter issues which might result in failure.

Our DKG is designed in a way that doesn't allow for inconsistent results: either it finishes correctly for every peer, or it fails.

This is asafety feature: you don't want to deposit an Ethereum distributed validator that not every operator is able to participate to, or even reach threshold for.

The most common source of issues lies in the network stack: if any of the peer's Internet connection glitches substantially, the DKG will fail.

Charon's DKG doesn't allow peer reconnection once the process is started, but it does allow for re-connections before that.

When you see the following message:

14:08:34.505 INFO dkg Waiting to connect to all peers... this means your Charon instance is waiting for all the other cluster peers to start their DKG process: at this stage, peers can disconnect and reconnect at will, the DKG process will still continue.

A log line will confirm the connection of a new peer:

14:08:34.523 INFO dkg Connected to peer 1 of 3 {"peer": "fantastic-adult"} 14:08:34.529 INFO dkg Connected to peer 2 of 3 {"peer": "crazy-bunch"} 14:08:34.673 INFO dkg Connected to peer 3 of 3 {"peer": "considerate-park"} As soon as all the peers are connected, this message will be shown:

14:08:34.924 INFO dkg All peers connected, starting DKG ceremony Past this stageno disconnections are allowed , andall peers must leave their terminals open in order for the DKG process to complete: this is a synchronous phase, and every peer is required in order to reach completion.

If for some reason the DKG process fails, you would see error logs that resemble this:

14:28:46.691 ERRO cmd Fatal error: sync step: p2p connection failed, please retry DKG: context canceled As the error message suggests, the DKG process needs to be retried.

Cleaning up the.charon

directory

One cannot simply retry the DKG process: Charon refuses to overwrite any runtime file in order to avoid inconsistencies and private key loss.

When attempting to re-run a DKG with an unclean data directory -- which is either.charon or what was specified with the--data-dir CLI parameter -- this is the error that will be shown:

14:44:13.448 ERRO cmd Fatal error: data directory not clean, cannot continue {"disallowed_entity": "cluster-lock.json", "data-dir": "/compose/node0"} Thedisallowed_entity field lists all the files that Charon refuses to overwrite, whiledata-dir is the full path of the runtime directory the DKG process is using.

In order to retry the DKG process one must delete the following entities, if present:

- validator_keys
- directory
- cluster-lock.json
- file
- deposit-data.json
- filecaution
- Thecharon-enr-private-key
- filemust be preserved
- , failure to do so requires the DKG process to be restarted from the beginning by creating a new cluster definition.
- If you're doing a DKG with a custom cluster definition - for example, create withcharon create dkg
- rather than the Obol Launchpad - you can re-use the same file.

Once this process has been completed, the cluster operators can retry a DKG.

Further debugging

If for some reason the DKG process fails again, node operators are adviced to reach out to the Obol team by opening

anissue , detailing what troubleshooting steps were taken and providingdebug logs .

To enable debug logs first clean up the Charon data directory as explained inthe previous paragraph , then run your DKG command by appending--log-level=debug at the end.

In order for the Obol team to debug your issue as quickly and precisely as possible please provide full logs in textual form, not through screenshots or display photos.

Providing complete logs is particularly important, since it allows the team to reconstruct precisely what happened.Edit this page Previous Centralization risks and mitigation Next Introduction