

URL support for all client options

| Metadata | Value |
|----------|-------------|
| Date | 2021-07-21 |
| Author | philpennock |
| Status | Deprecated |
| Tags | deprecated |

Deprecation Note

We discussed this among client authors and felt that we would not implement this for, among others, these reasons:

- Server lists holds many server urls but the client has just one set of options, UX allowing options on each seem sub optimal
- These parameters could expose sensitive information in logs and more
- Getting exact parity between all the clients would be hard, we would need to pick specific list of supported options but it would be forever churn to support more on all clients
- It would be difficult to always do the right thing wrt priority, in some cases you would want the URL to set defaults, in others you would want the URL to override settings

Overall we recognise this would be useful but decided against it for now.

Motivation

TBD

Overview

NATS URLs should be able to encode all information required to connect to a NATS server in a useful manner, except perhaps the contents of the CA certificate. This URL encoding should be consistent across all client languages, and be fully documented.

Making explicit comma-separated lists of URLs, vs of hostnames within a URL, and ensuring that is compatible across all clients is included, plus order randomization.

Anything tuning connection behavior, which might be used as an option on establishing a connection, should be specifiable in a URL. Anything which doesn't fit into authority information in the URL should probably be "query parameters", `?opt1=foo&opt2=bar` with the documentation establishing the option names and the behavior for unrecognized values for each option. Unrecognized options should be ignored. It's possible that some options should be `#fragopt1=foo&opt2=bar` instead and we should clearly define where we draw the line. Eg, "if it's not sent to the server but is reconnect timing information, it should be `#fragment`" or "for consistency we use `?query` for them all".

Things which should be configurable include, but are not limited to:

- OCSP checking status
- JetStream Domain
- Various timeouts
- TLS verification level (if we support anything other than verify always)
- Server TLS cert pinning per `hex(sha256(spki))`