Configure TLS

You can enable communications via TLS/SSL by setting"tls": "STRICT" .

If the value is set to"OFF" , the rest of the SSL configuration is ignored.

Important If using TLS, update the hostname of the node to usehttps instead ofhttp . TLS configuration { "sslConfig": { "tls": "[Authentication mode : OFF,STRICT]", "sslConfigType": "[Possible values: SERVER_ONLY, CLIENT_ONLY, SERVER_AND_CLIENT]",

// server options "serverTrustMode": "[Possible values: CA, TOFU, WHITELIST, CA_OR_TOFU, NONE]", "serverKeyStore": "[Path to server keystore]", "serverKeyStorePassword": "[Password required for server KeyStore]", "serverTrustStore": "[Server trust store path]", "serverTrustStorePassword": "[Password required for server trust store]", "serverTlsKeyPath": "[Path to server TLS key path]", "serverTlsCertificatePath": "[Path to server TLS cert path]", "serverTrustCertificates": ["[Array of truststore certificates if no truststore is defined.]"],

// client options "clientTrustMode": "[Possible values: CA, TOFU, WHITELIST, CA_OR_TOFU, NONE]", "clientKeyStore": "[Path to client keystore. The keystore that is used when communicating to other nodes.]", "clientKeyStorePassword": "[Password required for client KeyStore]", "clientTrustStore": "[Path to client TrustStore]", "clientTrustStorePassword": "[Password required for client trust store]", "clientTlsKeyPath": "[Path to client TLS Key]", "clientTlsCertificatePath": "[Path to client TLS cert]", "clientTrustCertificates": ["[Array of truststore certificates if no truststore is defined.]"],

"knownClientsFile": "[TLS known clients file for the server. This contains the fingerprints of public keys of other nodes that are allowed to connect to this one.]", "knownServersFile": "[TLS known servers file for the client. This contains the fingerprints of public keys of other nodes that this node has encountered.]", "generateKeyStoreIfNotExisted": "[boolean]", "environmentVariablePrefix": "[Prefix to uniquely identify environment variables for this particular server ssl configuration]", "clientAuth": "[Configure if SSL needs client authentication - boolean - default is true]" } } Tip To enable one-way SSL, setclientAuth tofalse . When TLS/SSL is enabled, each node must havecertificates andkeys defined for both client-side and server-side. You can define these in multiple ways, and in the following order of precedence:

1. Secured and unsecured.jks
2. (Java keystore) format files.
3. - serverKeyStore
4. - ,serverKeyStorePassword
5. - ,serverTrustStore
6. - ,serverTrustStorePassword
7. - clientKeyStore
8. - ,clientKeyStorePassword
9. - ,clientTrustStore
10. - ,clientTrustStorePassword
11. .pem
12. format certificate and key files.
13. - serverTlsKeyPath
14. - ,serverTlsCertificatePath
15. - ,serverTrustCertificates
16. - clientTlsKeyPath
17. - ,clientTlsCertificatePath
18. - ,clientTrustCertificates
19. .pem format configuration
20. "sslConfig" : {
21. "tls" : "STRICT",
22. "generateKeyStoreIfNotExisted" : "false",
23. "sslConfigType" : "SERVER_AND_CLIENT",
24. "serverTlsKeyPath" : "server-key.pem",

25. "serverTlsCertificatePath" : "server-cert.pem",
26. "serverTrustCertificates" : ["server-trust.pem"]
27. "serverTrustMode" : "CA",
28. "clientTlsKeyPath" : "client-key.pem",
29. "clientTlsCertificatePath" : "client-cert.pem",
30. "clientTrustCertificates" : ["client-trust.pem"]
31. "clientTrustMode" : "TOFU",
32. "knownClientsFile" : "knownClients",
33. "knownServersFile" : "knownServers"
34. }

Configuration type

When configuring TLS, use either the client configuration options or server configuration options depending on yousserver configuration . You can also definesslConfigType to limit TLS to only the appropriate options.

Server configuration TLS Configuration sslConfigType valueP2P Server and client SERVER_AND_CLIENT ThirdParty Server options only SERVER_ONLY Q2T Server options only SERVER_ONLY ENCLAVE Server and client SERVER_AND_CLIENT

Keystores

Passwords

You can provide passwords for secured.jks keystores in multiple ways, and in the following order of precedence:

1. Prefixed environment variables.
2.
    - _TESSERA_SERVER_KEYSTORE_PWD
3.
    - ,_TESSERA_SERVER_TRUSTSTORE_PWD
4.
    - _TESSERA_CLIENT_KEYSTORE_PWD
5.
    - ,_TESSERA_CLIENT_TRUSTSTORE_PWD
6. These are only applied to the servers with the correspondingenvironmentVariablePrefix
7. value defined in their configuration. This allows, for example, a P2P and ADMIN server to be configured with different prefixes,P2P
8. andADMIN
9. . Different keystores can then be used for each server and the individual passwords provided withP2P_<...>
10. andADMIN_<...>
11. .
12. Configuration file.
13.
    - serverKeyStorePassword
14.
    - ,serverTrustStorePassword
15.
    - clientKeyStorePassword
16.
    - ,clientTrustStorePassword
17. Global environment variables.
18.
    - TESSERA_SERVER_KEYSTORE_PWD
19.
    - ,TESSERA_SERVER_TRUSTSTORE_PWD
20.
    - TESSERA_CLIENT_KEYSTORE_PWD
21.
    - ,TESSERA_CLIENT_TRUSTSTORE_PWD
22. These are applied to all server configurations defined in the configuration file.

::: info

If a P2P and ADMIN server are both configured with TLS then the values set for the global environment variables are used for both. These values are ignored if the passwords are also provided in the configuration file or as prefixed environment variables.

:::

Generate keystores

If keystores don't already exist, Tessera can generate.jks files for use with non-CAtrust modes .

If you setgenerateKeyStoreIfNotExisted totrue , Tessera checks whether files already exist at the paths provided in theserverKeyStore andclientKeyStore configuration values. If the files don't exist:

1. Tessera generates new keystores and saves them at theserverKeyStore
2. andclientKeyStore
3. paths.
4. Tessera secures the keystores using the correspondingpasswords
5. if they're provided.

Trust modes

You must specify the trust mode for both client and server. Multiple trust modes are supportedTOFU ,WHITELIST ,CA ,CA_OR_TOFU , andNONE .

note If you use TLS on multiple endpoints (for example, P2P and Q2T) and run everything on localhost (or a single machine), then you must use differentknownClients andknownServers files for the different endpoints.

TOFU (Trust-on-first-use)

Only the first node that connects identifying as a certain host is allowed to connect as the same host in the future. When connecting for the first time, the host and its certificate are added toknownClientsFile (for server), orknownServersFile (for client). These files are generated if they don't already exist, using the values specified inknownClientsFile andknownServersFile .

TOFU trust mode configuration "sslConfig" : { "tls" : "STRICT", "generateKeyStoreIfNotExisted" : "true", "sslConfigType" : "SERVER_AND_CLIENT", "serverKeyStore" : "server-keystore", "serverKeyStorePassword" : "tessera", "serverTrustMode" : "TOFU", "clientKeyStore" : "client-keystore", "clientKeyStorePassword" : "tessera", "clientTrustMode" : "TOFU", "knownClientsFile" : "knownClients", "knownServersFile" : "knownServers" }

WHITELIST

Only nodes that have previously connected to this node and have been added to theknownClients file are allowed to connect. Similarly, this node can only make connections to nodes that have been added to theknownServers file. This trust mode doesn't add new entries to theknownClients orknownServers files.

With this trust mode, you must provide the allowlist (whitelist) files (knownClientsFile andknownServersFile).

WHITELIST trust mode configuration "sslConfig" : { "tls" : "STRICT", "generateKeyStoreIfNotExisted" : "true", "sslConfigType" : "SERVER_AND_CLIENT", "serverKeyStore" : "server-keystore", "serverKeyStorePassword" : "tessera", "serverTrustMode" : "WHITELIST", "clientKeyStore" : "client-keystore", "clientKeyStorePassword" : "tessera", "clientTrustMode" : "WHITELIST", "knownClientsFile" : "knownClients", "knownServersFile" : "knownServers" }

CA

Only nodes with a valid certificate and chain of trust are allowed to connect. For this trust mode, you must provide trust stores that contain a list of trust certificates.

CA trust mode configuration "sslConfig" : { "tls" : "STRICT", "generateKeyStoreIfNotExisted" : "false", //You can't generate trust stores when using CA "sslConfigType" : "SERVER_AND_CLIENT", "serverKeyStore" : "server-keystore", "serverKeyStorePassword" : "tessera", "serverTrustStore" : "server-truststore", "serverTrustStorePassword" : "tessera", "serverTrustMode" : "CA", "clientKeyStore" : "client-keystore", "clientKeyStorePassword" : "tessera", "clientTrustStore" : "client-truststore", "clientTrustStorePassword" : "tessera", "clientTrustMode" : "CA", "knownClientsFile" : "knownClients", "knownServersFile" : "knownServers" } Edit this page Last updatedonOct 9, 2023 bydependabot[bot]Previous Peer discovery Next Database