

vDRA

- Support for MongoDB 8.0 Version in vDRA, on page 1
- TLS 1.3 support, on page 3

Support for MongoDB 8.0 Version in vDRA

Feature Summary and Revision History

Table 1: Summary Data

Applicable Product(s) or Functional Area	vDRA
Applicable Platform(s)	Not Applicable
Default Setting	Enabled – Configuration Required
Related Changes in This Release	Not Applicable
Related Documentation	Not Applicable

Table 2: Revision History

Revision Details	Release
First introduced	26.2.0

Feature Description

This release provides support for MongoDB version 8.0.26

Upgrade, Migrate, and Backward Compatibility Considerations

- **Supported DRA Releases for Upgrading to 8.0**: Supported DRA releases for upgrading to version 8.0 include vDRA 25.1, 25.2, and 26.1, which use MongoDB version 7.0. You can upgrade these to vDRA 26.2, which uses MongoDB version 8.0.

- **Un Supported DRA Releases for Upgrading to 8.0**: Any DRA version prior to DRA 24.2 (mongo 6.0) like DRA 24.1 (mongo 5.0), DRA 23.1/23.2(mongo 4.4), and previous versions of DRA doesn't support direct upgrade to DRA 25.1 (mongo version 7.0).

  Refer the link for upgrading the replica set to 7.0.



Note

- Upgrading to MongoDB 8.0 is supported only from MongoDB 7.0. If you are running the 6.0 series, you must upgrade to 7.0 before upgrading to 8.0.

Mongo Java Driver: Current DRA Version 26.2.0 supports mongo java driver 3.12.9.

Prerequisite for upgrading to 26.2.0 from 25.1.0 or 26.1.0

The following are the common prerequisites for both upgrade and downgrade:

- Run the following CLI before upgrade:

```
#database genericfcvcheck 7.0
```



Note Make sure to run the above CLI before upgrade and / or downgrade on all sites.

- Specify any one of the CLI options:
  - **Set**: This option checks and sets FCV only on primary.



Note We recommend using the **Set** option first and then **Check** to make sure that FCV is replicated on primary members. Upgrade/downgrade should not be triggered if any error is found in the above CLI or FCV is not replicated on secondary members. Make sure to resolve the CLI error, rerun the CLI, and then only proceed for upgrade or downgrade.

  - **Check**: This option only checks FCV on all members (primary, secondary, and arbiter).

- Run the following CLI before upgrade:

```
#database dwccheck
```



Note CLI automatically takes care of the defaultWriteConcern version on all databases.

- Specify any one of the CLI options:
  - **Set**: This option checks and sets dwc on primary members.

Note We recommend using the **Set** option first and then **Check** to make sure that DWC is replicated on primary members. Upgrade/downgrade should not be triggered if any error is found in the above CLI or DWC is not replicated on secondary members. Make sure to resolve the CLI error, rerun the CLI, and then only proceed for upgrade or downgrade.

- **Check**: This option only checks dwc on all members.
- **(set/check) << set**
  - **Set**: This option checks and sets defaultWriteConcern.
  - **Check**: This option only checks defaultWriteConcern on all members(primary/secondary).

Upgrade to 26.2..0

1. Run the prerequisite steps.
2. Follow the standard documented procedure for upgrade.

Downgrade from 26.2.0

1. Run the steps mentioned in the prerequisite section.
2. Follow the standard documented procedure for downgrade.

TLS 1.3 support

TLS 1.3 Diameter connections

A TLS 1.3 Diameter connection is a security protocol enhancement that

- enables mutual TLS authentication between Cisco DRA nodes and Diameter peers,
- uses hardened cipher suites and handshake privacy to reduce exposure to network threats, and
- preserves interoperability by automatically falling back to TLS 1.2 as required.

Cisco DRA automatically prefers TLS 1.3 when supported by the peer, requiring no changes in Policy Builder configuration or CLI certificate management. Certificates that match TLS 1.2 criteria continue to operate for both TLS 1.2 and TLS 1.3. Certificate validation includes checks for key strength, signature algorithms, and extensions like Subject Alternative Name (SAN) and Extended Key Usage (EKU).

Table 3: Feature History Table

Feature Name	Release Information	Feature Description
TLS 1.3 support for Diameter connections in Cisco DRA	Release 26.2	You gain stronger security and reduced handshake latency for Diameter connections by enabling TLS 1.3 and mutual TLS in Cisco DRA. The system automatically negotiates TLS 1.3 with peers and falls back to TLS 1.2 if needed, allowing you to secure connections without changing existing workflows or certificate procedures.

Technical attributes and interoperability considerations for TLS 1.3 Diameter connections

The following attributes distinguish TLS 1.3 Diameter connections in Cisco DRA:

- TLS 1.3 reduces handshake latency (one round trip) and eliminates protocol renegotiation and legacy handshake messages.
- Only five cipher suites are supported in TLS 1.3, focusing on AEAD modes and SHA-256 or SHA-384 hashes.
- The peer monitoring GUI displays the negotiated TLS version for each connection.
- Automatic fallback to TLS 1.2 ensures continued operation if the peer does not support TLS 1.3; connections below minimum TLS version are rejected.

The cipher suite and certificate requirements for TLS 1.3 can be compared to TLS 1.2 as follows:

Table 4: Comparison of key attributes for TLS 1.2 and TLS 1.3 Diameter connections

Attributes	TLS 1.2	TLS 1.3
Supported cipher suites	Dozens, including CBC and legacy suites	Five, all AEAD with SHA-256/384
Handshake round trips	Two	One
Certificate requirements	Allows weaker keys and SHA-1 signatures	Requires RSA 2048-bit or ECDSA P-256/384 and SHA-256 or stronger

Certificate parameter compatibility for TLS 1.3 Diameter connections:

Table 5: Certificate attribute support for TLS 1.2 and TLS 1.3 Diameter connections

Certificate attribute	TLS 1.2	TLS 1.3
RSA 2048-bit key	Yes	Yes
RSA less than 2048-bit	May work	No
SHA-1 signature	May work	No

Certificate attribute	TLS 1.2	TLS 1.3
Subject Alternative Name extension	Not required	Required
Extended Key Usage (EKU)	Optional	Required

Note The peer monitoring GUI provides visibility into TLS version negotiation for each connection. Always check certificate details before deployment to avoid compatibility issues.

Best practice for TLS 1.3 support for Diameter connections

Ensure you issue and select certificates that meet Cisco DRA Diameter TLS 1.3 requirements. Certificates must use RSA 2048-bit or larger keys or ECDSA P-256/384, SHA-256 or stronger signature algorithms, and must include a Subject Alternative Name (SAN) extension. Define Extended Key Usage (EKU) attributes for TLS Web Server Authentication and TLS Web Client Authentication for mutual TLS. Avoid deprecated parameters.

- Replace certificates with SHA-1 signatures, RSA keys below 2048-bit, missing SAN, or incorrect EKU, since TLS 1.3 rejects them.
- Regenerate certificates after upgrades or configuration changes to maintain compliance and security.

Enable TLS 1.3 for Diameter connections and keep TLS 1.2 as fallback during migration. Monitor certificate status using Cisco DRA KPI `tls_cert_validity` and the peer monitoring GUI. After updating certificates, restart or re-establish peer connections. Keep backup copies of certificates and keys in secure locations.

- Do not allow connections with expired, deprecated, or non-compliant certificates.
- Regularly review certificate expiry and status to minimize downtime and maximize security.

Follow these certificate and operational practices for all Diameter peer connections secured by TLS 1.3 or mutual TLS in Cisco DRA. Apply recommendations to new deployments and during migration from TLS 1.2.

Strict certificate compliance and protocol selection prevent connection failures and security vulnerabilities. Modern cryptographic standards keep your network secure and interoperable across TLS versions.

You maintain secure, reliable Diameter connections, reduce certificate-related errors, and enable seamless migration between TLS versions.

Validate certificates using CLI commands like `openssl x509 -in cert.crt -noout -text | grep "Signature Algorithm"`, `openssl x509 -in cert.crt -noout -text | grep -A2 "Subject Alternative Name"`, and `openssl x509 -in cert.crt -noout -dates`. Use Policy Builder GUI or CLI for TLS configuration and verify certificates before activation.

How TLS 1.3 support for Diameter connections works

Cisco DRA (Diameter Routing Agent) supports TLS 1.3 for peer connections, enhancing security and maintaining compatibility with TLS 1.2. This process explains how secure connections are established and operated between DRA and Diameter peers.

- DRA nodes use encrypted channels and negotiate TLS versions, preferring TLS 1.3 whenever possible.

- DRA falls back to TLS 1.2 if the peer does not support TLS 1.3, ensuring interoperability.
- Certificate requirements and mutual TLS settings are enforced based on the negotiated protocol.

Summary

The key components involved in the process are:

- DRA software: Negotiates TLS protocol versions, validates certificates, and manages encryption and decryption for Diameter sessions.
- Policy Builder and CLI tools: Provide configuration and certificate management to enable TLS features.
- Peer Diameter nodes: Participate in TLS negotiation, present certificates, and exchange encrypted traffic with DRA.
- Peer monitoring GUI: Displays TLS status and certificate information for connections.

Secure connections are established by negotiating the highest supported TLS version, validating certificates, and completing handshakes using secure cipher suites. DRA falls back to TLS 1.2 or rejects connections if peer certificates do not meet compliance.

Workflow

The process involves these stages:

1. Connection initiation and protocol negotiation (performed by DRA and peer devices)
   - DRA initiates a connection or receives a request, then sends a ClientHello message listing supported TLS versions and ciphers.
   - The peer responds with a ServerHello message, selecting the highest mutual TLS version (typically TLS 1.3).

   If both devices support TLS 1.3, handshake messages use encrypted channels and approved cipher suites.

   When TLS 1.3 is negotiated, the connection is secured using advanced protocols.
2. Key exchange and session security
   - DRA and the peer use Elliptic Curve Diffie-Hellman Ephemeral (ECDHE) to derive session keys and enable forward secrecy.

   The key exchange protects session confidentiality even if certificate keys are compromised in the future.

   Session encryption keys are established for Diameter communication.
3. Certificate exchange and validation
   - DRA validates the peer certificate for TLS 1.3 requirements: RSA 2048-bit or ECDSA P-256/P-384, SHA-256 or stronger, Subject Alternative Name, and Extended Key Usage attributes.
   - If mutual TLS is enabled, both sides present certificates and validation is performed according to compliance settings.

   Noncompliant certificates (such as weak keys or missing SAN) cause connection failure or trigger fallback to TLS 1.2 if allowed.

Only properly validated certificates enable TLS 1.3 connectivity.

4. TLS handshake completion and Diameter data exchange
   - Successful certificate validation and key exchange conclude the handshake and begin secure Diameter data transfer after the Finished message.
   - TLS 1.3 disables renegotiation to improve protocol safety.

   Diameter messages are transmitted securely. Any certificate or configuration changes require reconnection.

   The connection operates using the negotiated TLS 1.3 parameters and cipher suite.
5. Connection monitoring and compatibility management
   - The DRA peer monitoring GUI shows TLS version and certificate status for each peer connection.
   - DRA falls back to TLS 1.2 if TLS 1.3 is not supported by the peer and configuration allows, using compatible ciphers and certificate specifications.
   - Connections using versions below the minimum (such as TLS 1.1) are rejected.

   Operational tools confirm security status and help detect misconfiguration or expired certificates.

   Only secure and compliant connections are maintained, and others are flagged or dropped based on policy.

Result

Cisco DRA's support for TLS 1.3 delivers robust encryption and forward secrecy for Diameter connections, ensuring operational continuity and maintaining compatibility with existing configuration settings.

What's next

For configuration instructions, see the process on configuring TLS certificates for Diameter connections.

Configure TLS 1.3 for Diameter connections

Configure Cisco DRA to enable TLS 1.3 for Diameter connections, improving encryption, handshake privacy, and forward secrecy.

- Secure communication between DRA nodes and Diameter peers using industry-standard TLS protocols.
- Maintain support for TLS 1.2 to allow migration and ensure interoperability with existing peers.

Cisco DRA supports TLS 1.3 and mutual TLS (mTLS) for Diameter peer connections, strengthening security and interoperability with TLS 1.2 peers.

All certificates must meet TLS 1.3 requirements to ensure secure connections. Peer monitoring tools display the negotiated TLS version.

- Use hardened cipher suites and certificates with appropriate key strength and attributes.
- The DRA CLI and Policy Builder GUI allow certificate import and protocol enablement.

Before you begin

Complete these prerequisites before configuring TLS 1.3 support for Diameter connections.

- Administrator access to Cisco DRA CLI and Policy Builder GUI.

- Certificate and private key files meeting TLS 1.3 criteria: RSA 2048-bit minimum or ECDSA P-256/P-384, SHA-256 or stronger, Subject Alternative Name (SAN), valid Extended Key Usage (EKU), and non-expired certificate.
- Certificate files must import without errors.

Follow these steps to configure TLS 1.3 support for Diameter connections:

Procedure

Step 1 Import the TLS certificate and private key into Cisco DRA using the CLI.

Example:

```
dra-tls cert import <certificate file> <private key file>
dra-tls cert import certificate.pem private.pem
```

Certificate import enables TLS for Diameter peer connections.

- Ensure the certificate is PEM-formatted and includes the SAN extension.
- Certificate import fails if the file does not comply with TLS 1.3 requirements.

Table 6: Certificate Import Conditions

If...	Then...
The certificate is expired or about to expire	Replace it with a valid certificate before import.
The certificate key size is less than RSA 2048-bit or SAN is missing	DRA rejects the certificate and TLS 1.3 connections cannot be established.
The certificate signature is SHA-1	Upgrade the certificate to SHA-256 or stronger for TLS 1.3 compatibility.

The certificate and key are imported successfully for TLS-secured Diameter connections.

Step 2 Verify certificate compatibility with TLS 1.3 using validation commands if required.

Example:

```
openssl x509 -in cert.crt -noout -text | grep "Signature Algorithm"
openssl x509 -in cert.crt -noout -text | grep "Public-Key"
openssl x509 -in cert.crt -noout -text | grep -A2 "Subject Alternative Name"
openssl x509 -in cert.crt -noout -dates
openssl x509 -in cert.crt -noout -text | grep -A2 "Extended Key Usage"
```

Validating certificate parameters confirms TLS 1.3 handshake and encryption support.

- Signature algorithm must be SHA-256 or stronger.
- RSA keys must be 2048 bits or higher, ECDSA must be P-256 or P-384.
- SAN should include DNS or IP entries.
- EKU must include "TLS Web Server Authentication" and "TLS Web Client Authentication" for mTLS.

Table 7: Certificate Validation Results

If...	Then...
Signature algorithm is weaker than SHA-256	Regenerate the certificate using SHA-256 or stronger.
Key size is less than RSA 2048-bit or incorrect EC curve	Obtain a new certificate with the correct parameters.
SAN or EKU is missing	Add the required fields to the certificate.

Certificate meets TLS 1.3 and mTLS requirements. Proceed to protocol enablement.

Step 3 Enable TLS or mutual TLS in the Policy Builder GUI for the Diameter interface.

Protocol enablement applies to the Diameter interface configuration.

- TLS enables server certificate validation by the client.
- mTLS enables mutual certificate validation by both server and client.
- No changes are needed for Policy Builder or CLI for TLS 1.2 deployments.

Table 8: Protocol Selection in GUI

If...	Then...
"TLS" is selected for the interface	DRA encrypts Diameter connections using TLS for that interface.
"mTLS" is selected for the interface	DRA requires mutual authentication (both server and client certificates checked).
No protocol is selected for the interface	Diameter connections remain unencrypted.

TLS or mTLS is enabled for the Diameter interface and ready for secure connections.

Step 4 Save the configuration and apply changes.

Saving ensures changes take effect on the next peer connection handshake.

- New settings apply when Diameter peers reconnect.
- Re-establishing connections updates peer security profiles.

Table 9: Configuration Save and Effect

If...	Then...
Configuration is not saved	New settings are not applied.
Peer reconnection does not occur	Changes may not take effect until the next handshake.
Configuration is saved	TLS 1.3 and mTLS features become active on new Diameter connections.

Diameter peer interfaces use TLS 1.3 or mTLS for security after saving the configuration.

Step 5 Verify that TLS 1.3 is being used for Diameter peer connections.

Verification confirms secure operation and correct protocol negotiation.

- Peer monitoring GUI shows the negotiated TLS version for each connection.
- KPI `tls_cert_validity` reports certificate expiration status.
- Secure connections display protocol version and status in logs.

Table 10: Connection and Certificate Validation

If...	Then...
Peer monitoring GUI shows TLS 1.3 for connections	Configuration is successful and secure communication is established.
KPI `tls_cert_validity` reports certificate is expired or invalid	Replace certificates and reconfigure if needed.
Connections are rejected or TLS 1.3 not negotiated	Check certificate parameters, protocol support, and configuration settings.

Diameter peer connections are secured with TLS 1.3 and all certificates are valid and compatible.

Diameter connections in Cisco DRA are secured using TLS 1.3 or mTLS, with enhanced encryption, privacy, and forward secrecy. Compatibility with TLS 1.2 peers is preserved.

What to do next

Monitor and maintain TLS configuration and certificates for ongoing secure operations.

- Check certificate expiration and rotate certificates before expiry.
- Review peer monitoring for TLS version and connection health.
- Maintain backup copies of certificates and keys in secure storage.
- Update certificates or configurations if peers report failures or rejected connections.