



SAP HANA Cloud, SAP HANA Database Security Guide

Generated on: 2024-11-26 04:01:11 GMT+0000

SAP HANA Cloud, SAP HANA Database | QRC 3/2024

PUBLIC

Original content: https://help.sap.com/docs/HANA_CLOUD_DATABASE/c82f8d6a84c147f8b78bf6416dae7290?locale=en-US&state=PRODUCTION&version=2024_3_QRC

Warning

This document has been generated from the SAP Help Portal and is an incomplete version of the official SAP product documentation. The information included in custom documentation may not reflect the arrangement of topics in the SAP Help Portal, and may be missing important aspects and/or correlations to other topics. For this reason, it is not for productive use.

For more information, please visit the <https://help.sap.com/docs/disclaimer>.

SAP HANA Cloud, **SAP HANA database** instances can be accessed using the standard SAP HANA client interfaces only through secure connections on the SQL port using the SAP HANA SQL command **network protocol through TCP/IP**.

2

- TLS_ECDHE_RSA_WITH_AES128_GCM_SHA256
- TLS_ECDHE_RSA_WITH_AES256_GCM_SHA384
- TLS_ECDHE_ECDSA_WITH_AES128_GCM_SHA256
- TLS_ECDHE_ECDSA_WITH_AES256_GCM_SHA384

The following cipher suites are enabled for TLS 1.3:

- TLS_AES_256_GCM_SHA384
- TLS_AES_128_GCM_SHA256

The following elliptic curves are enabled:

- EC_P384
- EC_P521
- EC_P256
- EC_X25519

Related Information

[Server Certificate Authentication \(SAP HANA Client Interface Programming Reference\)](#)

[How to Use the Client Interfaces with SAP HANA Cloud \(SAP HANA Client Interface Programming Reference\)](#)

<https://www.digicert.com/digicert-root-certificates.htm> ➔

[SAP Note 3327214](#) ➔

[SAP Note 3399573](#) ➔

Prerequisites for Secure SAP HANA Client Connections

The Transport Layer Security (TLS) protocol is used to secure SAP HANA client connections to SAP HANA Cloud, SAP HANA database instances. A number of prerequisites must be fulfilled client side.

- **SQLDBC**

To secure the connection to SAP HANA database instances, the SAP HANA client must use the SAP Common Crypto Library (CommonCryptoLib), Microsoft CryptoAPI, or OpenSSL. Connections with OpenSSL require an SAP HANA client version of 2.4.167 or higher, and OpenSSL versions 1.0.1, 1.0.2, 1.1.0, and 1.1.1 are supported for SNI connections.

The following SAP HANA client drivers are also based on SQLDBC: ADO.NET, .NET Core, Go, ODBC, Python, Node.js, and Ruby.

i Note

To connect to SAP HANA database instances using the Microsoft CryptoAPI, at least Microsoft Windows 7 and Microsoft Windows 2008 R2 are required. Microsoft Windows does not support OpenSSL connections.

- **JDBC**

Secure JDBC connections to SAP HANA database instances require a SAP HANA JDBC client version of 2.4.67 or higher and for SNI connections at least a JVM 8. The SAP HANA JDBC 2.4.67 driver is included with version 2.4.167 and later of the SAP HANA client.

i Note



JDBC uses the TLS implementation from the JVM.

- The SAP HANA client must contain the root certificate of the SAP HANA server instance.

i **Note**

If you encounter issues with the encrypted SQL connection, then verify that the trust store used by the SAP HANA client contains both the root certificate 'DigiCert Global Root CA' and 'DigiCert Global Root G5'. For `hdbsql` connections, the connection string must either explicitly reference the trust store that contains this root certificate or you can configure the SAP Common Crypto Library to get the certificate information from `$SECUDIR/sapcli.pse`. For JDBC-based connections, this is not necessary because the Java VM contains the required root certificate.

Related Information

- [Connect to SAP HANA Cloud via JDBC](#)
- [Connect to SAP HANA Cloud via ODBC](#)
- [SAP Note 3327214](#) 
- [SAP Note 3399573](#) 

Client-Side TLS Connection Properties (ODBC)

For ODBC-based connections, the configuration properties and their names are the same as the server parameters with the addition of the `encrypt` property, which initiates a TLS-secured connection.

The following table lists the configuration properties that are used to configure TLS for ODBC client access.

For more information about other ODBC connection properties and how to set them, see the *SAP HANA Client Interfaces Programming Reference*.

Property	Value	Default	Description
<code>encrypt</code>	Boolean	FALSE	Enables or disables . The server chooses the highest available.
<code>sslCryptoProvider</code>	{ <code>commoncrypto</code> <code>openssl</code> <code>mscrypto</code> }	<div>1. <code>commoncrypto</code></div> <div>2. <code>openssl/mscrypto</code></div>	<div>Specifies the cryptographic library provider used for . If you specify a value for this property, then you must also explicitly specify paths in both the <code>sslKeyStore</code> and <code>sslTrustStore</code> properties to avoid configuration issues.</div> <div>If <code>CommonCryptoLib</code> is not available, <code>OpenSSL</code> is used by default in Linux environments, and <code>msCrypto</code> in Microsoft Windows environments. <code>OpenSSL</code> is not available on Microsoft Windows.</div> <div>i Note</div> <div>Check the <code>client</code> folder of the installation package to see if <code>COMMONCRYPTOLIB.TGZ</code> is present. If not, you can download <code>CommonCryptoLib</code> separately. For instructions on how to download <code>CommonCryptoLib</code>, see "Download and Install SAP Common Crypto</div>

Property	Value	Default	Description
			Library" in the <i>SAP HANA Client Installation and Update Guide</i> .
sslHostNameInCertificate	<string>	Empty	<p>Writes the host name used to verify server's identity.</p> <p>The host name specified here verifies the identity of the server instead of the host name with which the connection was established.</p> <p>For example, in a single-host system, if a connection is established from a client on the same host as the server, then a mismatch would arise between the host named in the certificate (actual host name) and the host used to establish the connection (localhost).</p> <p>If you specify * as the host name, then the server's host name is not validated. Other wildcards are not permitted.</p>
sslKeyStore	<p><file> <PEM-encoded-identity></p> <p>msCrypto: MY Root Trust CA</p>	<p>\$SECUDIR/sapcli.pse (CommonCryptoLib)</p> <p>\$HOME/.ssl/key.pem (OpenSSL)</p> <p>MY (msCrypto)</p>	<p>Specifies the path to the keystore file that contains the client's identity. An identity is used for mutual TLS authentication and consists of the client's private key, the client's certificate, and, optionally, the certificate of the signing authority that signed the client's certificate. If you are using CommonCryptoLib, then the PSE file must also contain the trust store (for example, the server's public certificates) and the sslTrustStore property should be empty. If you are using CommonCryptoLib, but not doing mutual TLS authentication, then the PSE file contains only the trust store.</p> <p>If you are using CommonCryptoLib, then use the SAPGENPSE tool (installed with CommonCryptoLib) to create the client PSE (sapcli.pse) and generate the client's keys. You must also import the server's public certificates into sapcli.pse. Typically, this is the root certificate or the certificate of the certification authority that signed the server's public certificates. See <i>SAP Note 1718944 (SAP HANA DB: Securing External SQL Communication (CommonCryptoLib))</i>.</p> <p>If you are using OpenSSL, use OpenSSL tools to create the required keystore file.</p> <p>Alternatively, the PEM-encoded client identity can be provided directly as a</p>

Property	Value	Default	Description
			string. Use the SAPGENPSE tool to export the contents of a PSE file in PEM format. When you provide the client identity as a string, use the sslTrustStore property to provide the PEM-encoded trust store as a string.
sslTrustStore	<file> <PEM-encoded-trust-store> msCrypto: MY Root Trust CA	\$HOME/.ssl/trust.pem (OpenSSL) sapcli.pse (CommonCryptoLib) MY (msCrypto)	Specifies the path to a trust store file that contains the server's public certificates if using OpenSSL. Typically, the trust store contains the root certificate or the certificate of the certification authority that signed the server's public certificates. If you are using the cryptographic library CommonCryptoLib or msCrypto, leave this property empty. Alternatively, the PEM-encoded client identity can be provided directly as a string. Use the SAPGENPSE tool to export the contents of a PSE file in PEM format. When you provide the client identity as a string, use the sslTrustStore property to provide the PEM-encoded trust store as a string.
sslSNIHostname	<string>	Empty	Specifies the name of the host that is attempting to connect at the start of the TLS handshaking process.
sslValidateCertificate	Boolean	TRUE	Specifies whether to validate the server's certificate.

Related Information

[SAP HANA Client Interface Programming Reference](#)

Client-Side TLS Connection Properties (JDBC)

For clients connecting via the JDBC interface, TLS is configured using connection properties.

The following table lists the connection properties that can be used to configure TLS for JDBC client access.

For more information about other JDBC connection properties and how to set them, see the *SAP HANA Client Interfaces Programming Reference* in *Related Information* below.

Property	Value	Default	Description
encrypt	Boolean	FALSE	Enables or disables .
hostNameInCertificate	<string>	Empty	Writes the host name used to verify server's identity.

Property	Value	Default	Description
			<p>The host name specified here is used to verify the identity of the server instead of the host name with which the connection was established.</p> <p>For example, in a single-host system, if a connection is established from a client on the same host as the server, a mismatch would arise between the host named in the certificate (actual host name) and the host used to establish the connection (localhost).</p> <p>i Note</p> <p>If you specify * as the host name, this property has no effect. Other wildcards are not permitted.</p>
keyStore	<file> <store name>	<VM default>	Specifies the location of the Java keystore.
keyStorePassword	<password>	<VM default>	Specifies the password used to access the private key from the keystore file.
keyStoreType	<JKS> <PKCS12>	<VM default>	Specifies the Java keystore file format.
sniHostname	<string>	Empty	Specifies the name of the host that is attempting to connect at the start of the TLS handshaking process.
sslKeyStore	<string>	Empty	Specifies an alternative to the keyStore connection property. This property allows you to specify the contents of the keystore file as a string. The certificates and private keys must be PEM-encoded, and the private keys must be in PKCS8 format.
sslTrustStore	<string>	Empty	Specifies an alternative to the trustStore connection property. This property allows you to specify the contents of the truststore file as a string. The certificates must be PEM-encoded.
trustStore	<file> <store name>	<VM default>	<p>Specifies the path to the trust store file that contains the server's public certificate(s).</p> <p>Typically, the trust store contains the root certificate or the certificate of the certification authority that signed the server's certificate(s).</p>
trustStorePassword	<password>	<VM default>	Specifies the password used to access the trust store file.

Related Information

[SAP HANA Client Interface Programming Reference](#)

SOCKS Proxy Communication Protocol

SOCKS proxy is a communication protocol that uses a proxy server to exchange network packets between a client and server.

SOCKS proxy is an insecure network communication protocol.

You can connect to a SOCKS proxy server either by using SAP HANA HDBSQL or by using the SAP HANA client's ODBC driver.

SAP HANA HDBSQL

Specify the `-proxyhost <hostname>` option. You can also specify additional, optional SOCKS proxy hdbsql options.

SAP HANA Client ODBC driver

Specify the `PROXY_HOST <hostname>` connection property. You can also specify additional, optional SOCKS proxy ODBC connection properties.

Related Information

[SAP HANA HDBSQL Options](#)