Securing Automic Automation Database Connections

Connecting to PostgreSQL 12 Database

Version 1.1

Contents

Chapter 1: Introduction

PostgreSQL is one of the databases supported by Automic Automation for classical installations and deployments to Kubernetes clusters. Many Cloud providers offer managed database services on platforms like Azure, Google Cloud Platform, and AWS; ensuring a secure connection to the database is no longer optional.

This document does not replace the Automic documentation or a basic understanding of the TLS/SSL protocol and other relevant concepts, such as private/public keys and certificates.

The following is only an example of configuring the Database Server and Automic to secure connections between the Automation Engine and an PostgreSQL 12 database. Other setups might be a better fit based on your needs.

.

Chapter 2: Configure TLS/SSL for the PostgreSQL Database Server

This guide uses a self-signed server certificate to secure the connection between the database and Automation Engine, but it is also possible to configure certificates signed by a Certificate Authority (CA).

Client certificates are not configured in this example, but can be set up as described in the PostgreSQL documentation.

2.1 Create TLS/SSL self-signed certificate for the Database Server

Create a private key and a self-signed certificate with OpenSSL:

```
$ openssl req -new -x509 -days 365 -nodes -text -out server.crt -keyout server.key \
  -subj "/CN=mydbserver"
```

If hostname verification is required, the Common Name (CN) in the certificate must match the hostname/domain/address that the Automic system will use to connect to the Database Server.

Include additional hostnames or domain names as Subject Alternative Names (SANs) while creating the certificate.

2.2 Configure the PostgreSQL Server to use TLS/SSL

The path to the private key and certificate needs to be configured in the Database Server configuration file and the server must be restarted.

postgresql.conf:

```
# - SSL -
ssl = on
#ssl_ca_file = ''
ssl_cert_file = 'server.crt'
#ssl_crl_file = ''
ssl_key_file = 'server.key'
```

In order to ensure that all clients connect using TLS/SSL, the default entries in pg_ba.conf related to host have to be removed (for example “host all all 127.0.0.1”) and hostssl ones configured instead.

pg_hba.conf:

```
# TYPE    DATABASE    USER      ADDRESS       METHOD
…
host      all         all       127.0.0.1
hostssl   all         all       0.0.0.0/0     md5
hostssl   all         all       ::/0          md5
```

Chapter 3: Configure TLS/SSL for Automic

PostgreSQL allows configuring different security levels on the client-side as documented here

To enable server hostname verification, the most restrictive option, the sslmode needs to be set to verify-full. Since the server certificate is self-signed, the root certificate used by the client for authenticating the server is the same as server.crt and is set with the sslrootcert parameter.

The secure connection is established on the Automic server-side between the ODBC/JDBC drivers used by the worker and communication processes and the Database Server.

In the example below, verify-full is configured, so the database host in the server ini file needs to match the Common Name (CN) in the certificate for the hostname verification to succeed.

ucsrv.ini:

```
[ODBC]
sqlDriverConnect=ODBCVAR=NNJNIORP,host=mydbserver port=5432 dbname=ae_main user=ae
password=oab connect_timeout=10 client_encoding=LATIN9 sslmode=verify-full
sslrootcert=C:\\Users\\ob685245\\AppData\\Roaming\\pgAdmin\\server.crt

[JDBC]
sqlDriverConnect=jdbc:postgresql://mydbserver:5432/ae_main?sslmode=verify-full&sslrootce
rt=C:\\Users\\ob685245\\AppData\\Roaming\\pgAdmin\\server.crt
```

If the path to the certificate does not exist or the certificate is not valid, errors similar to the ones below will be written to the log:

```
20210512/082612.763 - 35     U00003545 UCUDB: Opening database ...
20210512/082612.885 - 35              Connection error:
20210512/082612.886 - 35     U00003611 DB OPEN executed. Return Code = '1'
20210512/082612.886 - 35     U00003590 UCUDB - DB error: '08006', 'Could not open SSL root certificate file
C:\Users\ob685245\AppData\Roaming\postgresql\root.crt.', '0', 'org.postgresql.util.PSQLException'
20210512/082612.887 - 35     U00032031 Error when connecting to Database

20210512/083402.437 - 1      U00003545 UCUDB: Opening database ...
20210512/083402.708 - 1              Connection error:
20210512/083402.708 - 1      U00003611 DB OPEN executed. Return Code = '1'
20210512/083402.709 - 1      U00003590 UCUDB - DB error: '08006', 'SSL error: PKIX path building failed:
sun.security.provider.certpath.SunCertPathBuilderException: unable to find valid certification path to requested
target', '0', 'org.postgresql.util.PSQLException'
20210512/083402.710 - 1      U00032031 Error when connecting to Database
```


BROADCOM
SOFTWARE