aidev mongoDB MONITORING PLUG-IN FOR ORACLE ENTERPRISE MANAGER

SSL Configuration: an example

July 2016

This document details a walkthrough example of SSL configuration in an EM managed mongoDB environment.

SSL certificates are used to enforce certificate based security and network encryption.

ROOT KEY AND CA CERTIFICATE CREATION:

This will be used to sign the mongoDB and agent certificates.

For the mongoDB instance the trusted CA maps to the parameter *sslCAFile*.

```
mkdir /etc/ssl/mongo

cd /etc/ssl/mongo

openssl genrsa -out rootCA.key 4096
```

We now have a root key:

rootCA.key

Generate the root certificate (only required if you need a new one):

```
openssl req -x509 -new -nodes -key rootCA.key -days 3650 -out
rootCA.pem
```

```
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:UK
State or Province Name (full name) [Some-State]:
Locality Name (eg, city) []:
Organization Name (eg, company) [Internet Widgits Pty Ltd]:AIDEV UK
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:AIDEV ROOT CA
Email Address []:info@aidev.uk
```

We also now have a root certificate and key:

rootCA.key
rootCA.pem

MONGO KEY AND CERTIFICATE CREATION

This will be used by the mongoDB instance and maps to the parameter *sslPEMKeyFile*.

IMPORTANT: PLEASE ENSURE THAT THE COMMON NAME MATCHES THE HOSTNAME OR IP ADDRESS OF THE MONGODB HOST.

```
openssl genrsa -out mongoDB.key 4096
```

```
openssl req -new -key mongoDB.key -out mongoDB.csr
```

```
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:GB
State or Province Name (full name) [Some-State]:
Locality Name (eg, city) []:
Organization Name (eg, company) [Internet Widgits Pty Ltd]:AIDEV UK
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:192.168.0.51
Email Address []:info@aidev.uk

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:welcome1
An optional company name []:
```

```
openssl x509 -req -in mongoDB.csr -CA rootCA.pem -CAkey rootCA.key
-CAcreateserial -out mongoDB.crt -days 3650
```

```
Signature ok
subject=/C=GB/ST=Some-State/O=AIDEV
UK/CN=osboxes/emailAddress=info@aidev.uk
Getting CA Private Key
```

For the mongoDB side, we now have:

mongoDB.crt
mongoDB.csr
mongoDB.key

AGENT KEY AND CERTIFICATE CREATION

This will be used by the EM agent.

```
openssl genrsa -out agent.key 4096
```

```
openssl req -new -key agent.key -out agent.csr
```

```
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:GB
State or Province Name (full name) [Some-State]:
Locality Name (eg, city) []:
Organization Name (eg, company) [Internet Widgits Pty Ltd]:AIDEV UK
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:emcc.example.com
Email Address []:info@aidev.uk

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:welcome1
An optional company name []:
```

```
openssl x509 -req -in agent.csr -CA rootCA.pem -CAkey rootCA.key -CAcreateserial -out agent.crt -days 3650
```

```
Signature ok
subject=/C=GB/ST=Some-State/O=AIDEV UK/CN=emcc.example.com/emailAddress=info@aidev.uk
Getting CA Private Key
```

For the agent side we now have:

agent.crt
agent.csr
agent.key

CONFIGURE MONGOD

Create the pem files for the agent and mongoDB:

cat mongoDB.key mongoDB.crt > mongoDB.pem

cat agent.key agent.crt > agent.pem

IMPORTANT: ensure the following files are chmod 600 by the mongoDB operating system user:

mongoDB.pem
rootCA.pem

Start mongo:

```
/root/mongo/mongodb-linux-x86_64-ubuntu1404-3.0.4/bin/mongod  \
--fork --rest --port 28001 \
--dbpath /data_new/db/ssl_01 \
--logpath /data_new/db/logs/mongod_ssl_01.log \
--auth \
--sslPEMKeyFile /etc/ssl/mongo/mongoDB.pem \
--sslMode requireSSL \
--sslCAFile /etc/ssl/mongo/rootCA.pem
```

Alternatively if using a configuration file add in the following:

```
sslPEMKeyFile = /data_new/db/keys/ssl_01.pem
sslCAFile =/etc/ssl/mongo/rootCA.pem
sslMode = requireSSL
#sslMode = allowSSL
#sslMode = preferSSL
```

Please refer to the official mongoDB documentation for a detailed description of the sslMode options.

TRUSTSTORE CONFIGURATION

Create the truststore - this is required on the agent side:

IMPORTANT: use *welcome1* as the password for this agent internal store

Firstly, to ensure the private key and certificates are imported we need to convert the agent cert/key to pkcs12:

```
openssl pkcs12 -export -in agent.crt -inkey agent.key -out agent.p12 -name agentKey -CAfile rootCA.pem -caname root
```

We can then import using keytool:

```
keytool -importkeystore -deststorepass welcome1 -destkeystore truststore.ts \
 -srckeystore agent.p12 -srcstoretype PKCS12 -srcstorepass welcome1 -alias agentKey
```

Then add in the root CA:

```
keytool -importcert -trustcacerts -file /etc/ssl/mongo/rootCA.pem -alias rootCA -keystore ./truststore.ts -storepass welcome1
```

Now verify that all is ok:

```
keytool -list -keystore ./truststore.ts
```

```
Enter keystore password:

Keystore type: JKS
Keystore provider: SUN

Your keystore contains 2 entries

rootca, 04-Sep-2015, trustedCertEntry,
Certificate fingerprint (MD5):
4F:3E:51:A8:B7:A2:74:F8:47:0B:82:4B:71:F8:FC:4D
agentkey, 04-Sep-2015, PrivateKeyEntry,
Certificate fingerprint (MD5):
36:CD:0A:D0:D4:6E:BC:F4:F8:BF:5A:81:2C:62:5C:F3
```

AGENT AND TARGET CONFIGURATION

Copy the following files to the agent host-

truststore.ts

agent.pem

rootCA.pem

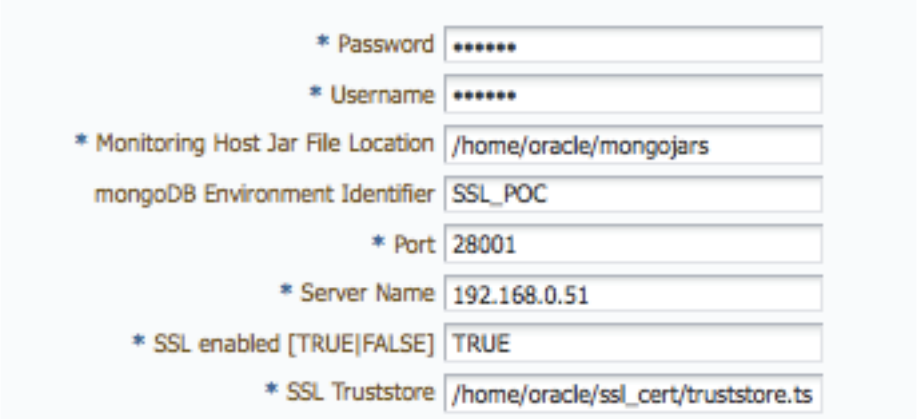
IMPORTANT: chmod 600 each of the above files to ensure only the agent operating system user can access them.

In this example we will hold these in /home/oracle/ssl_cert:

```
-rw-------  1 oracle dba    5154 Sep  6 03:30 agent.pem
-rw-------  1 oracle dba    2025 Sep  6 03:38 rootCA.pem
-rw-------  1 oracle dba    5363 Sep  6 02:33 truststore.ts
```

Targets within EM require the ‘SSL enabled’ and ‘SSL Truststore’ parameters to be set. This will allow SSL configuration to function for agent monitoring.

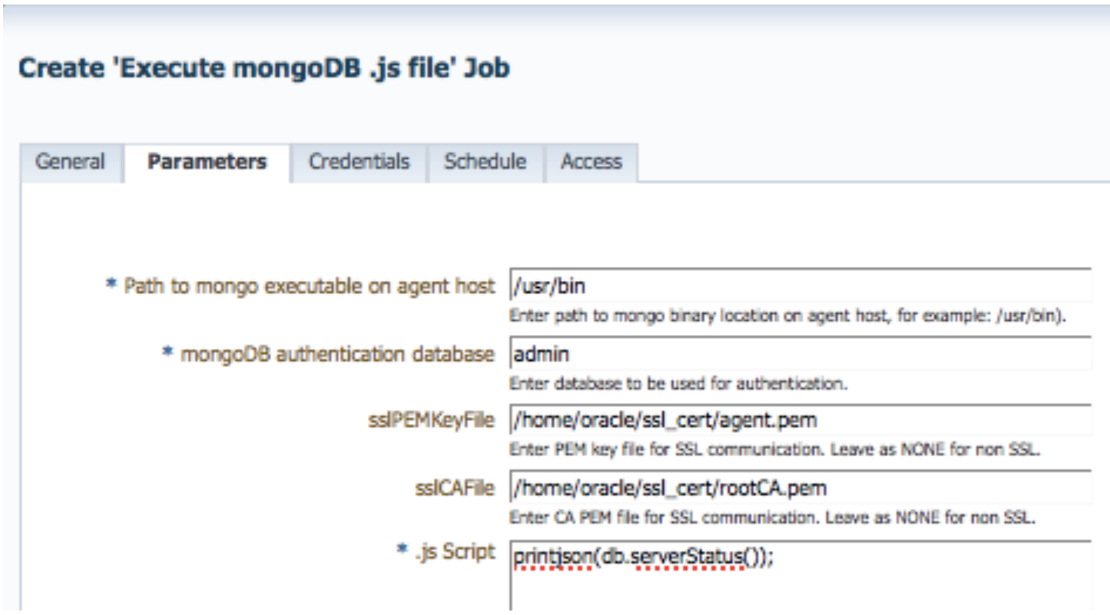
An example target configuration:



* Password	••••••
* Username	••••••
* Monitoring Host Jar File Location	/home/oracle/mongojars
mongoDB Environment Identifier	SSL_POC
* Port	28001
* Server Name	192.168.0.51
* SSL enabled [TRUE|FALSE]	TRUE
* SSL Truststore	/home/oracle/ssl_cert/truststore.ts

The mongoDB .js job execution feature requires the definition of rootCA, sslPemKeyFile and sslCAFile parameters.

An example .js job definition:



Create 'Execute mongoDB .js file' Job

* Path to mongo executable on agent host: /usr/bin
Enter path to mongo binary location on agent host, for example: /usr/bin).

* mongoDB authentication database: admin
Enter database to be used for authentication.

sslPEMKeyFile: /home/oracle/ssl_cert/agent.pem
Enter PEM key file for SSL communication. Leave as NONE for non SSL.

sslCAFile: /home/oracle/ssl_cert/rootCA.pem
Enter CA PEM file for SSL communication. Leave as NONE for non SSL.

* .js Script:

```
printjson(db.serverStatus());
```

aidev

NEED FURTHER INFORMATION? contact info@aidev.uk for more details on this product and how to join up with us.

Specifications and product offerings subject to change without notice.

Please ensure you have appropriate licenses for running this product. Aidev takes no responsibility whatsoever re. site licensing of Oracle EM, mongoDB or any other software used with this plugin – this responsibility lies solely with the Client.