#

# From oslo.db

#

# If True, SQLite uses synchronous mode. (boolean value)

#sqlite\_synchronous = true

# The back end to use for the database. (string value)

# Deprecated group/name - [DEFAULT]/db\_backend

#backend = sqlalchemy

# The SQLAlchemy connection string to use to connect to the database.

# (string value)

# Deprecated group/name - [DEFAULT]/sql\_connection

# Deprecated group/name - [DATABASE]/sql\_connection

# Deprecated group/name - [sql]/connection

#connection = <None>

{%- set connection\_x509\_ssl\_option = '' %}

{%- if \_data.get('x509',{}).get('enabled',False) %}

{%- set connection\_x509\_ssl\_option = '&ssl\_ca=' ~ \_data.x509.get('ca\_file') ~ '&ssl\_cert=' ~ \_data.x509.get('cert\_file') ~ '&ssl\_key=' ~ \_data.x509.get('key\_file') %}

{%- elif \_data.get('ssl',{}).get('enabled',False) %}

{%- set connection\_x509\_ssl\_option = '&ssl\_ca=' ~ \_data.ssl.get('cacert\_file') %}

{%- endif %}

connection = {{ \_data.engine }}+pymysql://{{ \_data.user }}:{{ \_data.password }}@{{ \_data.host }}/{{ \_data.name }}?charset=utf8{{ connection\_x509\_ssl\_option|string }}

# The SQLAlchemy connection string to use to connect to the slave

# database. (string value)

#slave\_connection = <None>

# The SQL mode to be used for MySQL sessions. This option, including

# the default, overrides any server-set SQL mode. To use whatever SQL

# mode is set by the server configuration, set this to no value.

# Example: mysql\_sql\_mode= (string value)

#mysql\_sql\_mode = TRADITIONAL

# If True, transparently enables support for handling MySQL Cluster

# (NDB). (boolean value)

#mysql\_enable\_ndb = false

# Connections which have been present in the connection pool longer

# than this number of seconds will be replaced with a new one the next

# time they are checked out from the pool. (integer value)

# Deprecated group/name - [DATABASE]/idle\_timeout

# Deprecated group/name - [database]/idle\_timeout

# Deprecated group/name - [DEFAULT]/sql\_idle\_timeout

# Deprecated group/name - [DATABASE]/sql\_idle\_timeout

# Deprecated group/name - [sql]/idle\_timeout

#connection\_recycle\_time = 3600

# (obryndzii) we change default connection\_recycle\_time to 280 in order to fix numerous

# DBConnection errors in services until we implement this setting in reclass-system

{%- if \_data.get('connection\_recycle\_time', 280) %}

connection\_recycle\_time = {{ \_data.get('connection\_recycle\_time', 280) }}

{%- endif %}

# Minimum number of SQL connections to keep open in a pool. (integer

# value)

# Deprecated group/name - [DEFAULT]/sql\_min\_pool\_size

# Deprecated group/name - [DATABASE]/sql\_min\_pool\_size

#min\_pool\_size = 1

# Maximum number of SQL connections to keep open in a pool. Setting a

# value of 0 indicates no limit. (integer value)

# Deprecated group/name - [DEFAULT]/sql\_max\_pool\_size

# Deprecated group/name - [DATABASE]/sql\_max\_pool\_size

#max\_pool\_size = 5

{%- if \_data.get('max\_pool\_size', 10) %}

max\_pool\_size = {{ \_data.get('max\_pool\_size', 10) }}

{%- endif %}

# Maximum number of database connection retries during startup. Set to

# -1 to specify an infinite retry count. (integer value)

# Deprecated group/name - [DEFAULT]/sql\_max\_retries

# Deprecated group/name - [DATABASE]/sql\_max\_retries

#max\_retries = 10

{%- if \_data.get('max\_retries', -1) %}

max\_retries = {{ \_data.get('max\_retries', -1) }}

{%- endif %}

# Interval between retries of opening a SQL connection. (integer

# value)

# Deprecated group/name - [DEFAULT]/sql\_retry\_interval

# Deprecated group/name - [DATABASE]/reconnect\_interval

#retry\_interval = 10

# If set, use this value for max\_overflow with SQLAlchemy. (integer

# value)

# Deprecated group/name - [DEFAULT]/sql\_max\_overflow

# Deprecated group/name - [DATABASE]/sqlalchemy\_max\_overflow

#max\_overflow = 50

{%- if \_data.get('max\_overflow', 30) %}

max\_overflow = {{ \_data.get('max\_overflow', 30) }}

{%- endif %}

# Verbosity of SQL debugging information: 0=None, 100=Everything.

# (integer value)

# Minimum value: 0

# Maximum value: 100

# Deprecated group/name - [DEFAULT]/sql\_connection\_debug

#connection\_debug = 0

# Add Python stack traces to SQL as comment strings. (boolean value)

# Deprecated group/name - [DEFAULT]/sql\_connection\_trace

#connection\_trace = false

# If set, use this value for pool\_timeout with SQLAlchemy. (integer

# value)

# Deprecated group/name - [DATABASE]/sqlalchemy\_pool\_timeout

#pool\_timeout = <None>

# Enable the experimental use of database reconnect on connection

# lost. (boolean value)

#use\_db\_reconnect = false

# Seconds between retries of a database transaction. (integer value)

#db\_retry\_interval = 1

# If True, increases the interval between retries of a database

# operation up to db\_max\_retry\_interval. (boolean value)

#db\_inc\_retry\_interval = true

# If db\_inc\_retry\_interval is set, the maximum seconds between retries

# of a database operation. (integer value)

#db\_max\_retry\_interval = 10

# Maximum retries in case of connection error or deadlock error before

# error is raised. Set to -1 to specify an infinite retry count.

# (integer value)

#db\_max\_retries = 20

#

# From oslo.db.concurrency

#

# Enable the experimental use of thread pooling for all DB API calls

# (boolean value)

# Deprecated group/name - [DEFAULT]/dbapi\_use\_tpool

#use\_tpool = false