#

# From oslo.messaging

#

# Use durable queues in AMQP. (boolean value)

# Deprecated group/name - [DEFAULT]/amqp\_durable\_queues

# Deprecated group/name - [DEFAULT]/rabbit\_durable\_queues

#amqp\_durable\_queues = false

# Auto-delete queues in AMQP. (boolean value)

#amqp\_auto\_delete = false

# Connect over SSL. (boolean value)

# Deprecated group/name - [oslo\_messaging\_rabbit]/rabbit\_use\_ssl

#ssl = false

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

ssl = true

# SSL version to use (valid only if SSL enabled). Valid values are TLSv1 and

# SSLv23. SSLv2, SSLv3, TLSv1\_1, and TLSv1\_2 may be available on some

# distributions. (string value)

# Deprecated group/name - [oslo\_messaging\_rabbit]/kombu\_ssl\_version

#ssl\_version =

{%- if \_data.ssl.version is defined %}

ssl\_version = {{ \_data.ssl.version }}

{%- elif salt['grains.get']('pythonversion') > [2,7,8] %}

ssl\_version = TLSv1\_2

{%- endif %}

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

# SSL key file (valid only if SSL enabled). (string value)

# Deprecated group/name - [oslo\_messaging\_rabbit]/kombu\_ssl\_keyfile

#ssl\_key\_file =

ssl\_key\_file = {{ \_data.x509.key\_file }}

# SSL cert file (valid only if SSL enabled). (string value)

# Deprecated group/name - [oslo\_messaging\_rabbit]/kombu\_ssl\_certfile

#ssl\_cert\_file =

ssl\_cert\_file = {{ \_data.x509.cert\_file }}

# SSL certification authority file (valid only if SSL enabled). (string value)

# Deprecated group/name - [oslo\_messaging\_rabbit]/kombu\_ssl\_ca\_certs

#ssl\_ca\_file =

ssl\_ca\_file = {{ \_data.x509.ca\_file }}

{%- else %}

ssl\_ca\_file = {{ \_data.ssl.cacert\_file }}

{%- endif %}

{%- endif %}

# How long to wait before reconnecting in response to an AMQP consumer cancel

# notification. (floating point value)

#kombu\_reconnect\_delay = 1.0

{%- if \_data.kombu\_reconnect\_delay is defined %}

kombu\_reconnect\_delay = {{ \_data.kombu\_reconnect\_delay }}

{%- endif %}

# EXPERIMENTAL: Possible values are: gzip, bz2. If not set compression will not

# be used. This option may not be available in future versions. (string value)

#kombu\_compression = <None>

{%- if \_data.kombu\_compression is defined %}

kombu\_compression = {{ \_data.kombu\_compression }}

{%- endif %}

# How long to wait a missing client before abandoning to send it its replies.

# This value should not be longer than rpc\_response\_timeout. (integer value)

# Deprecated group/name - [oslo\_messaging\_rabbit]/kombu\_reconnect\_timeout

#kombu\_missing\_consumer\_retry\_timeout = 60

{%- if \_data.kombu\_missing\_consumer\_retry\_timeout is defined %}

kombu\_missing\_consumer\_retry\_timeout = {{ \_data.kombu\_missing\_consumer\_retry\_timeout }}

{%- endif %}

# Determines how the next RabbitMQ node is chosen in case the one we are

# currently connected to becomes unavailable. Takes effect only if more than

# one RabbitMQ node is provided in config. (string value)

# Possible values:

# round-robin - <No description provided>

# shuffle - <No description provided>

#kombu\_failover\_strategy = round-robin

{%- if \_data.kombu\_failover\_strategy is defined %}

kombu\_failover\_strategy = {{ \_data.kombu\_failover\_strategy }}

{%- endif %}

# DEPRECATED: The RabbitMQ broker address where a single node is used. (string

# value)

# This option is deprecated for removal.

# Its value may be silently ignored in the future.

# Reason: Replaced by [DEFAULT]/transport\_url

#rabbit\_host = localhost

# DEPRECATED: The RabbitMQ broker port where a single node is used. (port

# value)

# Minimum value: 0

# Maximum value: 65535

# This option is deprecated for removal.

# Its value may be silently ignored in the future.

# Reason: Replaced by [DEFAULT]/transport\_url

#rabbit\_port = 5672

# DEPRECATED: RabbitMQ HA cluster host:port pairs. (list value)

# This option is deprecated for removal.

# Its value may be silently ignored in the future.

# Reason: Replaced by [DEFAULT]/transport\_url

#rabbit\_hosts = $rabbit\_host:$rabbit\_port

# DEPRECATED: The RabbitMQ userid. (string value)

# This option is deprecated for removal.

# Its value may be silently ignored in the future.

# Reason: Replaced by [DEFAULT]/transport\_url

#rabbit\_userid = guest

# DEPRECATED: The RabbitMQ password. (string value)

# This option is deprecated for removal.

# Its value may be silently ignored in the future.

# Reason: Replaced by [DEFAULT]/transport\_url

#rabbit\_password = guest

# The RabbitMQ login method. (string value)

# Possible values:

# PLAIN - <No description provided>

# AMQPLAIN - <No description provided>

# RABBIT-CR-DEMO - <No description provided>

#rabbit\_login\_method = AMQPLAIN

# DEPRECATED: The RabbitMQ virtual host. (string value)

# This option is deprecated for removal.

# Its value may be silently ignored in the future.

# Reason: Replaced by [DEFAULT]/transport\_url

#rabbit\_virtual\_host = /

# How frequently to retry connecting with RabbitMQ. (integer value)

#rabbit\_retry\_interval = 1

{%- if \_data.rabbit\_retry\_interval is defined %}

rabbit\_retry\_interval = {{ \_data.rabbit\_retry\_interval }}

{%- endif %}

# How long to backoff for between retries when connecting to RabbitMQ. (integer

# value)

#rabbit\_retry\_backoff = 2

{%- if \_data.rabbit\_retry\_backoff is defined %}

rabbit\_retry\_backoff = {{ \_data.rabbit\_retry\_backoff }}

{%- endif %}

# Maximum interval of RabbitMQ connection retries. Default is 30 seconds.

# (integer value)

#rabbit\_interval\_max = 30

{%- if \_data.rabbit\_interval\_max is defined %}

rabbit\_interval\_max = {{ \_data.rabbit\_interval\_max }}

{%- endif %}

# DEPRECATED: Maximum number of RabbitMQ connection retries. Default is 0

# (infinite retry count). (integer value)

# This option is deprecated for removal.

# Its value may be silently ignored in the future.

#rabbit\_max\_retries = 0

{%- if \_data.rabbit\_max\_retries is defined %}

rabbit\_max\_retries = {{ \_data.rabbit\_max\_retries }}

{%- endif %}

# Try to use HA queues in RabbitMQ (x-ha-policy: all). If you change this

# option, you must wipe the RabbitMQ database. In RabbitMQ 3.0, queue mirroring

# is no longer controlled by the x-ha-policy argument when declaring a queue.

# If you just want to make sure that all queues (except those with auto-

# generated names) are mirrored across all nodes, run: "rabbitmqctl set\_policy

# HA '^(?!amq\.).\*' '{"ha-mode": "all"}' " (boolean value)

#rabbit\_ha\_queues = false

{%- if \_data.rabbit\_ha\_queues is defined %}

rabbit\_ha\_queues = {{ \_data.rabbit\_ha\_queues }}

{%- endif %}

# Positive integer representing duration in seconds for queue TTL (x-expires).

# Queues which are unused for the duration of the TTL are automatically

# deleted. The parameter affects only reply and fanout queues. (integer value)

# Minimum value: 1

#rabbit\_transient\_queues\_ttl = 1800

{%- if \_data.rabbit\_transient\_queues\_ttl is defined %}

rabbit\_transient\_queues\_ttl = {{ \_data.rabbit\_transient\_queues\_ttl }}

{%- endif %}

# Specifies the number of messages to prefetch. Setting to zero allows

# unlimited messages. (integer value)

#rabbit\_qos\_prefetch\_count = 0

{%- if \_data.rabbit\_qos\_prefetch\_count is defined %}

rabbit\_qos\_prefetch\_count = {{ \_data.rabbit\_qos\_prefetch\_count }}

{%- endif %}

# Number of seconds after which the Rabbit broker is considered down if

# heartbeat's keep-alive fails (0 disable the heartbeat). EXPERIMENTAL (integer

# value)

#heartbeat\_timeout\_threshold = 60

{%- if \_data.get('heartbeat\_timeout\_threshold', 0) %}

heartbeat\_timeout\_threshold = {{ \_data.get('heartbeat\_timeout\_threshold', 0) }}

{%- endif %}

# How often times during the heartbeat\_timeout\_threshold we check the

# heartbeat. (integer value)

#heartbeat\_rate = 2

{%- if \_data.heartbeat\_rate is defined %}

heartbeat\_rate = {{ \_data.heartbeat\_rate }}

{%- endif %}