



Memory Use

RabbitMQ can report on its own memory use, to let you see where your system is using memory.

Note that all measurements are somewhat approximate, based on values returned by the underlying Erlang virtual machine; however they should still be accurate enough to be useful.

You can obtain the memory use report by invoking `rabbitmqctl status`, or by looking at the node details page in the **management** web UI. Memory use is split into the following categories (which do not overlap):

Connections

This includes memory used by incoming connections and channels, and outgoing ones if the appropriate plugins are loaded. It also includes memory used by the SSL system.

Queues

Memory used by individual queue processes. Note that queues will swap their contents out to disc when under memory pressure. Message bodies do not show up here but in Binaries.

Plugins

Memory used by plugins (apart from the Erlang client which is counted under Connections, and the management database which is counted separately). Note that RabbitMQ will count some per-connection memory here for protocol plugins such as STOMP and MQTT.

Other process memory

Memory belonging to processes not counted above, and memory assigned to "processes" by the Erlang VM, but not to any one process. Memory that has recently been garbage collected can show up here briefly.

Mnesia

Mnesia keeps an in-memory copy of all its data (even on disc nodes). Typically this will only be large when there are a large number of queues, exchanges, bindings, users or virtual hosts.

Message store index

The default message store implementation keeps an in-memory index of all messages, including those paged out to disc.

Management database

The management database (if the management plugin is loaded). In a cluster, this will only be present on one node.

Other ETS tables

Other in-memory tables besides the three sets above. Note that due to a bug in current versions of the Erlang runtime, some memory will be counted under this heading for all tables, including the three sets above.

Binaries

Memory used by shared binary data in the Erlang VM. In-memory message bodies show up here.

Code

Memory used by code. Should be fairly constant.

Atoms

Memory used by atoms. Should be fairly constant.

Other system memory

Other memory used by Erlang. One contributor to this value is the number of available file descriptors.

In This Section

- ❖ **Server Documentation**
 - ❖ Configuration
 - ❖ SSL Support
 - ❖ Distributed RabbitMQ
 - ❖ Reliable Delivery
 - ❖ Clustering
 - ❖ High Availability
 - ❖ High Availability (pacemaker)
 - ❖ Access Control
 - ❖ SASL Authentication
 - ❖ Flow Control
- ❖ **Memory Use**
 - ❖ Firehose / Tracing
 - ❖ Manual Pages
 - ❖ Windows Quirks
- ❖ **Client Documentation**
- ❖ Plugins
- ❖ News
- ❖ Protocol
- ❖ Our Extensions
- ❖ Building
- ❖ Previous Releases
- ❖ License