

# Cheatsheet: RabbitMQ Monitoring (Management Plugin)

- Note:**
- If you are using the Management plugin version of RabbitMQ, please use Pages 1 and 2. If you are using the Prometheus plugin version of RabbitMQ to access metrics in their OpenMetrics format, please use Pages 3 and 4.
  - To execute the `curl` requests below, you must [enable the management plugin](#) and replace `user:password` with your own username and password
  - `localhost:15672` is the default location for version 3.0 and later. `%2F` is the URL-encoded name of the default vhost, `/`

## Exchange performance metrics—more info

METRIC DESCRIPTION	COMMAND
Messages published into an exchange	<code>curl -u user:password localhost:15672/api/exchanges/%2F&lt;exchange_name&gt;   jq .message_stats.publish_in</code>
Messages published out of an exchange	<code>curl -u user:password localhost:15672/api/exchanges/%2F&lt;exchange_name&gt;   jq .message_stats.publish_out</code>
Messages that can't be routed to an exchange	<code>curl -u user:password http://localhost:15672/api/channels/&lt;channel_name&gt;   jq .message_stats.return_unroutable</code>

## Node metrics—more info

METRIC DESCRIPTION	COMMAND
File descriptors used	<code>curl -u user:password http://localhost:15672/api/nodes/&lt;node_name&gt;   jq .fd_used</code>
File descriptors used as sockets	<code>curl -u user:password http://localhost:15672/api/nodes/&lt;node_name&gt;   jq .sockets_used</code>
Disk space free (bytes)	<code>curl -u user:password http://localhost:15672/api/nodes/&lt;node_name&gt;   jq .disk_free</code>
Memory used (bytes)	<code>curl -u user:password http://localhost:15672/api/nodes/&lt;node_name&gt;   jq .mem_used</code>

## Connection performance metrics—more info

METRIC DESCRIPTION	COMMAND
Data rates	<code>curl -u user:password http://localhost:15672/api/connections/&lt;connection_name&gt;   jq '.recv_oct_details, .send_oct_details'</code>

## Queue performance metrics—more info

METRIC DESCRIPTION	COMMAND
Queue depth	<code>curl -u user:password http://localhost:15672/api/queues/%2F&lt;queue_name&gt;   jq .messages</code>
Messages unacknowledged	<code>curl -u user:password http://localhost:15672/api/queues/%2F&lt;queue_name&gt;   jq .messages_unacknowledged</code>
Messages ready	<code>curl -u user:password http://localhost:15672/api/queues/%2F&lt;queue_name&gt;   jq .messages_ready</code>
Message rates	<code>curl -u user:password http://localhost:15672/api/queues/%2F&lt;queue_name&gt;   jq .message_stats</code>
Messages written to disk	<code>curl -u user:password http://localhost:15672/api/queues/%2F&lt;queue_name&gt;   jq .messages_persistent</code>
Message bytes written to disk	<code>curl -u user:password http://localhost:15672/api/queues/%2F&lt;queue_name&gt;   jq .message_bytes_persistent</code>
Message bytes in memory	<code>curl -u user:password http://localhost:15672/api/queues/%2F&lt;queue_name&gt;   jq .message_bytes_ram</code>
Number of consumers	<code>curl -u user:password http://localhost:15672/api/queues/%2F&lt;queue_name&gt;   jq .consumers</code>
Consumer utilization	<code>curl -u user:password http://localhost:15672/api/queues/%2F&lt;queue_name&gt;   jq .consumer_utilisation</code>

Collect these OOTB metrics with Datadog

START YOUR FREE TRIAL



- To get the names of your channels and connections, consult the management web server UI or query the `/api/channels` OR `/api/connections/` endpoints, e.g. `curl -u user:password localhost:15672/api/channels`
- While examples here use HTTP, you can configure the management API to use [HTTPS](#)
- Commands for retrieving metrics from the API parse JSON with `jq`, which you can find [here](#)

## Default directories and config locations (Linux)

RESOURCE	LOCATION
Configuration file	<code>/etc/rabbitmq/rabbitmq.conf</code>
Log file	<code>/var/log/rabbitmq/&lt;node_name&gt;.log</code>
Simple Authentication and Security Layer (SASL) log file	<code>/var/log/rabbitmq/&lt;node_name&gt;-sasl.log</code>
Database directory	<code>/var/lib/rabbitmq/mnesia/&lt;node_name&gt;/</code>

## Useful commands (Linux)

ACTION	COMMAND
Start the broker	<code>sudo rabbitmq-server</code>
Stop the broker	<code>sudo rabbitmqctl stop</code>
See if a node is running and check memory use and application versions	<code>sudo rabbitmqctl status</code>
Enable a plugin	<code>sudo rabbitmq-plugins enable &lt;plugin_name&gt;</code>
List available and enabled plugins	<code>sudo rabbitmq-plugins list</code>

## Items available to list with `rabbitmqctl`

Report a quick list of metrics with `sudo rabbitmqctl list-<item>`

```
users
vhosts
permissions
user_permissions
topic_permissions
user_topic_permissions
parameters
global_parameters
policies
operator_policies
vhost_limits
bindings
queues
exchanges
connections
channels
consumers
hashes
ciphers
amqp10_connections
mqtt_connections
stomp_connections
```

# Cheatsheet: RabbitMQ Monitoring with Datadog (Management Plugin)



## Note:

- While RabbitMQ looks for configuration files at their default locations, you will need to create the files yourself.
- The metrics on this page correspond to the metrics showcased via our out-of-the-box dashboard for this integration.



## 1. Overview

METRIC DESCRIPTION	DATADOG METRIC NAME
RabbitMQ status by host (each healthy host adds 1)	rabbitmq.aliveness
Integration status (each reporting host adds 1)	rabbitmq.status
Disk Alarms	rabbitmq.node.disk_alarm
Memory Alarms	rabbitmq.node.mem_alarm

## 2. Queue throughput

METRIC DESCRIPTION	DATADOG METRIC NAME
Average messages entering or leaving the queue per second	rabbitmq.queue.messages.rate
Average messages published per second	rabbitmq.queue.messages.publish.rate
Average messages delivered per second	rabbitmq.queue.messages.deliver.rate
Average messages acknowledged per second	rabbitmq.queue.messages.ack.rate

## 3. Queue consumption

METRIC DESCRIPTION	DATADOG METRIC NAME
Queues by depth	rabbitmq.queue.messages
Consumer utilization	rabbitmq.queue.consumer_utilisation

## 4. Resource utilization

METRIC DESCRIPTION	DATADOG METRIC NAME
Free Disk Space	rabbitmq.node.disk_free
Used Memory	rabbitmq.node.mem_used
File descriptors used	rabbitmq.node.fd_used

Monitor RabbitMQ Technology with Datadog Free

START YOUR FREE TRIAL

# Cheatsheet: RabbitMQ Monitoring (Prometheus Plugin)

**Note:**

- If you are using the Management plugin version of RabbitMQ, please use Pages 1 and 2. If you are using the Prometheus plugin version of RabbitMQ to access metrics in their OpenMetrics format, please use Pages 3 and 4.

Node metrics—more info	
METRIC DESCRIPTION	COMMAND
Open file descriptors	<code>curl http://localhost:15692/metrics/detailed?family=node_coarse_metrics   grep rabbitmq_detailed_process_open_fds</code>
Open TCP sockets	<code>curl http://localhost:15692/metrics/detailed?family=node_coarse_metrics   grep rabbitmq_detailed_process_open_tcp_sockets</code>
Disk space available (bytes)	<code>curl http://localhost:15692/metrics/detailed?family=node_coarse_metrics   grep rabbitmq_detailed_disk_space_available_bytes</code>
Memory used (bytes)	<code>curl http://localhost:15692/metrics/detailed?family=node_coarse_metrics   grep rabbitmq_detailed_process_resident_memory_bytes</code>
Erlang processes used	<code>curl http://localhost:15692/metrics/detailed?family=node_coarse_metrics   grep rabbitmq_detailed_erlang_processes_used</code>
Node uptime	<code>curl http://localhost:15692/metrics/detailed?family=node_metrics   grep rabbitmq_detailed_erlang_uptime_seconds</code>

Queue Metrics—more info	
METRIC DESCRIPTION	COMMAND
Queue depth	<code>curl http://localhost:15692/metrics/detailed?family=queue_coarse_metrics   grep rabbitmq_detailed_queue_messages</code>
Messages ready to be delivered to consumers	<code>curl http://localhost:15692/metrics/detailed?family=queue_coarse_metrics   grep rabbitmq_detailed_queue_messages_ready</code>
Messages delivered but not yet acknowledged	<code>curl http://localhost:15692/metrics/detailed?family=queue_coarse_metrics   grep rabbitmq_detailed_queue_messages_unacked</code>
Consumers on a queue	<code>curl http://localhost:15692/metrics/detailed?family=queue_consumer_count   grep rabbitmq_detailed_queue_consumers</code>
Consumer utilization	<code>curl http://localhost:15692/metrics/detailed?family=queue_metrics   grep rabbitmq_detailed_queue_consumer_utilisation</code>
Message bytes stored in memory	<code>curl http://localhost:15692/metrics/detailed?family=queue_metrics   grep rabbitmq_detailed_queue_messages_ram_bytes</code>
Message bytes paged out to disk	<code>curl http://localhost:15692/metrics/detailed?family=queue_metrics   grep rabbitmq_detailed_queue_messages_paged_out_bytes</code>

Connection Metrics—more info	
METRIC DESCRIPTION	COMMAND
Outgoing bytes	<code>curl http://localhost:15692/metrics/detailed?family=connection_coarse_metrics   grep rabbitmq_detailed_connection_outgoing_bytes_total</code>
Incoming bytes	<code>curl http://localhost:15692/metrics/detailed?family=connection_coarse_metrics   grep rabbitmq_detailed_connection_incoming_bytes_total</code>

Channel/Exchange Metrics—more info	
METRIC DESCRIPTION	COMMAND
Messages published into an exchange on a channel	<code>curl http://localhost:15692/metrics/detailed?family=channel_exchange_metrics   grep rabbitmq_detailed_channel_messages_published_total</code>
Total number of messages published as mandatory into an exchange and returned to the publisher as unroutable	<code>curl http://localhost:15692/metrics/detailed?family=channel_exchange_metrics   grep rabbitmq_detailed_channel_messages_unroutable_returned_total</code>
Total number of messages published as non-mandatory into an exchange and dropped as unroutable	<code>curl http://localhost:15692/metrics/detailed?family=channel_exchange_metrics   grep rabbitmq_detailed_channel_messages_unroutable_dropped_total</code>

Collect these OOTB metrics with Datadog

START YOUR FREE TRIAL

  
DATADOG

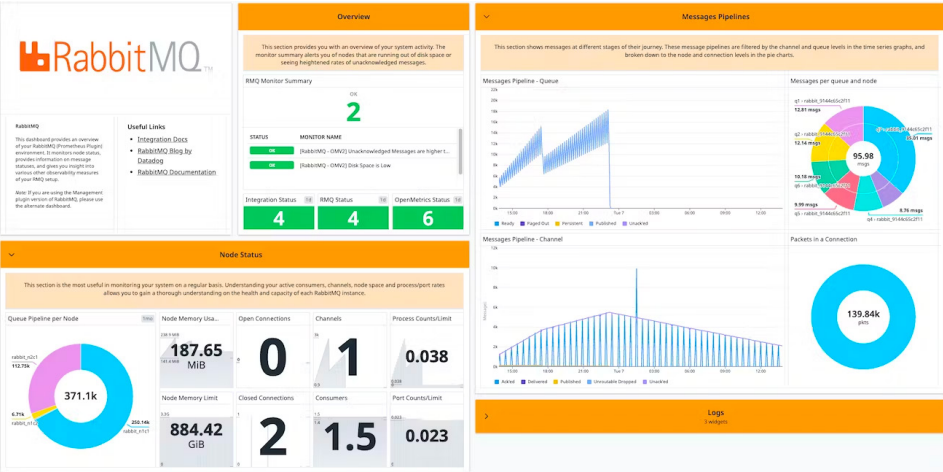
Default directories and config locations (Linux)	
RESOURCE	LOCATION
Configuration file	<code>/etc/rabbitmq/rabbitmq.conf</code>
Log file	<code>/var/log/rabbitmq/&lt;node_name&gt;.log</code>
Simple Authentication and Security Layer (SASL) log file	<code>/var/log/rabbitmq/&lt;node_name&gt;-sasl.log</code>
Database directory	<code>/var/lib/rabbitmq/mnesia/&lt;node_name&gt;/</code>

Useful commands (Linux)	
ACTION	COMMAND
Start the broker	<code>sudo rabbitmq-server</code>
Stop the broker	<code>sudo rabbitmqctl stop</code>
See if a node is running and check memory use and application versions	<code>sudo rabbitmqctl status</code>
Enable a plugin	<code>sudo rabbitmq-plugins enable &lt;plugin_name&gt;</code>
List available and enabled plugins	<code>sudo rabbitmq-plugins list</code>

# Cheatsheet: RabbitMQ Monitoring with Datadog (Prometheus Plugin)



- Note:**
- While RabbitMQ looks for configuration files at their default locations, you will need to create the files yourself.
  - The metrics on this page correspond to the metrics showcased via our out-of-the-box dashboard for this integration.



## 1. Overview

METRIC DESCRIPTION	DATADOG METRIC(S) NAME
RMQ Monitor Summary	(Search for your RMQ Monitors in the query bar)
OpenMetrics Status	Check Name: RabbitMQ - OpenMetrics endpoint health

## 2. Node Status

METRIC DESCRIPTION	DATADOG METRIC(S) NAME
Queue Pipeline per Node	rabbitmq.queues.created.count rabbitmq.queues.deleted.count rabbitmq.queues.declared.count rabbitmq.queues
Node Memory Usage	rabbitmq.process.resident_memory_bytes
Node Memory Limit	rabbitmq.resident_memory_limit_bytes
Open Connections	rabbitmq.connections.opened.count
Closed Connections	rabbitmq.connections.closed.count
Channels	rabbitmq.channels
Consumers	rabbitmq.global.consumers
Process Counts/Limit	Ratio between: rabbitmq.erlang.vm.process_count rabbitmq.erlang.vm.process_limit
Port Counts/Limit	Ratio between: rabbitmq.erlang.vm.port_count rabbitmq.erlang.vm.port_limit

## 3. Message Pipelines

METRIC DESCRIPTION	DATADOG METRIC(S) NAME
Message Pipeline - Queue	rabbitmq.queue.messages.ready rabbitmq.queue.messages.paged_out rabbitmq.queue.messages.persistent rabbitmq.queue.messages.published.count rabbitmq.queue.messages.unacked
Messages per queue and node	rabbitmq.queue.messages
Messages Pipeline - Channel	rabbitmq.channel.messages.delivered.ack.count rabbitmq.channel.messages.delivered.count rabbitmq.channel.messages.published.count rabbitmq.channel.messages.unroutable.dropped.count rabbitmq.channel.messages.unacked
Packets in a Connection	rabbitmq.connection.incoming_packets.count rabbitmq.connection.outgoing_packets.count rabbitmq.connection.pending_packets

## 4. Logs

METRIC DESCRIPTION	DATADOG METRIC(S) NAME
Counts per Log Status	Source:rabbitmq \$node_name Count * group by Status(status)
Error Logs for RabbitMQ	Source:rabbitmq ERROR

Monitor RabbitMQ Technology with Datadog Free

START YOUR FREE TRIAL