Cloud data management

# Cloud data management#

There are two concerns when managing data on Cloud:

• Memory usage: complex workflows processing large amounts of data can exceed n8n's memory limits. If this happens, the instance can crash and become inaccessible.

• Data storage: depending on your execution settings and volume, your n8n database can grow in size and run out of storage.

To avoid these issues, n8n recommends that you build your workflows with memory efficiency in mind, and don't save unnecessary data

## Memory limits on each Cloud plan#

Current plans:

• Trial: 320MiB RAM, 10 millicore CPU burstable

• Starter: 320MiB RAM, 10 millicore CPU burstable

• Pro-1 (10k executions): 640MiB RAM, 20 millicore CPU burstable

• Pro-2 (50k executions): 1280MiB RAM, 80 millicore CPU burstable

• Enterprise: 4096MiB RAM, 80 millicore CPU burstable

Legacy plans:

• Start: 320MiB RAM, 10 millicore CPU burstable

• Power: 1280MiB RAM, 80 millicore CPU burstable

n8n gives each instance up to 100GB of data storage.

## How to reduce memory consumption in your workflow#

The way you build workflows affects how much data they consume when executed. Although these guidelines aren't applicable to all cases, they provide a baseline of best practices to avoid exceeding instance memory.

• Split the data processed into smaller chunks. For example, instead of fetching 10,000 rows with each execution, process 200 rows with each execution.

• Avoid using the Code node where possible.

• Avoid manual executions when processing larger amounts of data.

• Split the workflow up into sub-workflows and ensure each sub-workflow returns a limited amount of data to its parent workflow.

Splitting the workflow might seem counter-intuitive at first as it usually requires adding at least two more nodes: the Loop Over Items node to split up the items into smaller batches and the Execute Workflow node to start the sub-workflow.

However, as long as your sub-workflow does the heavy lifting for each batch and then returns only a small result set to the main workflow, this reduces memory consumption. This is because the sub-workflow only holds the data for the current batch in memory, after which the memory is free again.

Note that n8n itself consumes memory to run. On average, the software alone uses around 180MiB RAM.

Interactions with the UI also consume memory. Playing around with the workflow UI while it performs heavy executions could also push the memory capacity over the limit.

## How to manage execution data on Cloud#

Execution data includes node data, parameters, variables, execution context, and binary data references. It's text-based.

Binary data is non-textual data that n8n can't represent as plain text. This is files and media such as images, documents, audio files, and videos. It's much larger than textual data.

If a workflow consumes a large amounts of data and is past testing stage, it's a good option to stop saving the successful executions.

There are two ways you can control how much execution data n8n stores in the database:

In the admin dashboard:

• From your workspace or editor, navigate to Admin Panel.

• Select Manage.

• In Executions to Save deselect the executions you don't want to log.

In your workflow settings:

• Select the Options menu.

• Select Settings. n8n opens the Workflow settings modal.

• Change Save successful production executions to Do not save.

## Cloud data pruning and out of memory incident prevention#

### Automatic data pruning#

n8n automatically prunes execution logs after a certain time or once you reach the max storage limit, whichever comes first. The pruning always happens from oldest to newest and the limits depend on your Could plan:

• Start and Starter plans: max 2500 executions saved and 7 days execution log retention;

• Pro and Power plans: max 25000 executions saved and 30 days execution log retention;

• Enterprise plan: max 50000 executions saved and unlimited execution log retention time.

### Manual data pruning#

Heavier executions and use cases can exceed database capacity despite the automatic pruning practices. In cases like this, n8n will manually prune data to protect instance stability.

• An alert system warns n8n if an instance is at 85% disk capacity.

• n8n prunes execution data. n8n does this by running a backup of the instance (workflows, users, credentials and execution data) and restoring it without execution data.

Due to the human steps in this process, the alert system isn't perfect. If warnings are triggered after hours or if data consumption rates are high, there might not be time to prune the data before the remaining disk space fills up.