Uptime Infrastructure Monitor Sizing Examples

Sizing varies based on the number of monitors per element, the type of objects monitored, and the method used to gather performance data. Uptime Infrastructure Monitor recommendations are based on an average of 2 or 3 monitors per element.

During Uptime Infrastructure Monitor's installation, one of three options was selected depending on the size of your monitored environment. The choice determined how certain resources were allocated, and subsequent hardware requirements:

Elements	Minimum RAM	Minimum CPU Type
< 200	8 GB	4-cores/vCPUs
201 - 1000	32 GB	8-cores/vCPUs
1001 - 5000	128 GB	24-cores/vCPUs

As a general rule of thumb when planning the allocated disk space, you should plan on allocating about 4 GB per monitored element. Note that per element usages are per year, unless you archive using old data.

Each sizing example ensures the underlying configurations (for example, for the bundled MySQL database and JRE) match the likely resource demands to generate reports, dashboards, and perform queries. The following summarizes these configurations:

Configuration Parameter	< 200 Elements	201 - 1000 Elements	1001 - 5000 Elements
MySQL buffer pool size	120 MB	2 GB	4 GB
MySQL log file size	20 MB	512 MB	1 GB
MySQL maximum open connections	151	201	301
Java heap size	1 GB	2 GB	4 GB
service threads	50	100	200
Data Collector maximum open connections	100	150	250
up.time Controller heap size	512 MB	1 GB	2 GB

Modifying the Sizing Examples

After initial installation, if you need to accommodate a larger number of monitored Elements, you can manually change the sizing examples using one of two methods:

The first, more direct, option is to individually modify parameters that make up a template. This allows you to deviate somewhat from the prescribed configuration values for a template:

Configuration Parameter	Configuration File and Location (relative to the Uptime Infrastructure Monitor directory)	Parameter Name	Default Sizing Values (< 200, 201 - 1000, 1001 - 5000)	

MySQL buffer pool size	`/mysql/my.ini`	`innodb_buffer_pool_size=`	`120M`
			`2G`
			`4G`	

MySQL log file size	`/mysql/my.ini`	`innodb_log_file_size=`	`20M`
			`512M`
			`1G`	

MySQL maximum open connections	`/mysql/my.ini`	`max_connections=`	`151`
			`201`
			`301`	

Java heap size	Linux:
	`/uptime.jncf`
	
	Windows:
	`\UptimeDataCollector.ini`
		Linux:
		`-Xmx<size>`
		
		Windows:
		`vm.heapsize.preferred=`
			Linux:
			`-Xmx1G -Xmx2G -Xmx4G`
				Windows:
				`1024m`
				`2048m`
				`4096m`

service threads	`/uptime.conf`	`serviceThreads=`	`50`
			`100`
			`200`	

Data Collector maximum open connections	`/uptime.conf`	`connectionPoolMaximum=`	`100`
			`150`
			`250`	

up.time Controller heap size	Linux:
	`/controller/service/start.sh`
	
	Windows:
	`\controller\service\UptimeController.ini`
		Linux:
		`-Xmx<size>`
		
		Windows:
		`vmarg.2=`
			`-Xmx512m`
			`-Xmx1024m`
			`-Xmx2048m`

The recommended option to change a sizing example is to use the sample configuration files that are found in the `<uptimeInstall>/sample` directory as a starting point. This option moves you to a different sizing example in the least amount of steps.

Modifying Sample Templates

Update the DataStore configuration:

1. Stop the DataStore service (uptime_datastore on Linux, or "up.time Data Store" on Windows)
2. Move the `ib_logfile0` and `ib_logfile1` files, found in the `<uptimeInstall>/datastore/data/` directory, to a backup location.
3. Back up the MySQL `my.ini` configuration file, which is found in the `<uptimeInstall>/mysql/` directory.
4. Copy the `<uptimeInstall>/sample/<size>/<os>/my.ini` template file to the `<uptimeInstall>/mysql/` directory, replacing the existing one.
5. Edit the `my.ini` file, replacing all `$VARIABLE$` instances with values that match your Uptime Infrastructure Monitor deployment (for example, `$DATASTORE_PORT$` and `$USER_INSTALL_DIR$`)
6. Start the DataStore service
   You can confirm the change was successful by referring to the `<uptimeInstall>/datastore/data/<hostname>.err` log. Look for output similar to the following:

```
140110 14:26:28 InnoDB: Initializing buffer pool, size = 2.0G
140110 14:26:29 InnoDB: Completed initialization of buffer pool
140110 14:26:29  InnoDB: Log file .\ib_logfile0 did not exist: new to be created
InnoDB: Setting log file .\ib_logfile0 size to 512 MB
InnoDB: Database physically writes the file full: wait...
InnoDB: Progress in MB: 100 200 300 400 500
140110 14:26:33  InnoDB: Log file .\ib_logfile1 did not exist: new to be created
InnoDB: Setting log file .\ib_logfile1 size to 512 MB
InnoDB: Database physically writes the file full: wait...
InnoDB: Progress in MB: 100 200 300 400 500
```

Update the up.time Data Collector configuration:

1. Back up the `uptime.conf` file, which is found in the `<uptimeInstall>/` directory.
2. Back up the Data Collector configuration file (`uptime.jcnf` on Linux, or `UptimeDataCollector.ini` on Windows), which is found in the `<uptimeInstall>/` directory.
3. Copy the `uptime.conf`, and `uptime.jcnf` or `UptimeDataCollector.ini` files from the `<uptimeInstall>/sample/<size>/<os>/` directories to the `<uptimeInstall>/` directory, replacing the existing ones.
4. Edit both files, replacing all `$VARIABLE$` instances with values that match your Uptime Infrastructure Monitor deployment (for example, the `$MS_STRING_FS$` classpath variable, and `$DATASTORE_HOST$`).
5. Restart the up.time data-collection service (uptime_core on Linux, or "up.time Data Collector" on Windows)

Update the up.time Controller configuration:

1. Back up the up.time Controller configuration file:
   - Linux: the `start.sh` script, which is found in the `<uptimeInstall>/controller/service/` directory
   - Windows: the `UptimeController.ini` configuration file, which is found in the `<uptimeInstall>\controller\service` directory
2. Copy the `start.sh` or `UptimeController.ini` file from the `<uptimeInstall>/sample/<size>/<os>/` directory to the `<uptimeInstall>/controller/service/` directory, replacing the existing one.
3. Edit the file, replacing all `$VARIABLE$` instances with values that match your Uptime Infrastructure Monitor deployment.
4. Restart the up.time Controller service (uptime_controller on Linux, or up.time Controller on Windows)

Update the up.time Web server:

1. Back up the `php.ini` up.time Web server configuration file, which is found in the `<uptimeInstall>/apache/conf/` directory.
2. Copy the `php.ini` file from the `<uptimeInstall>/sample/<size>/<os>/` directory to the `<uptimeInstall>/apache/conf/` directory, replacing the existing one.
3. Edit the file, replacing all `$VARIABLE$` instances with values that match your Uptime Infrastructure Monitor deployment (for example, `$USER_INSTALL_DIR$`).
4. Restart the up.time Web server (uptime_httpd on Linux, or "up.time Web Server" on Windows).