monit-docker Documentation

Release 0.0.47

Adrien Delle Cave

Sep 26, 2022

Contents

monit-docker is a free and open-source, we develop it to monitor container status or resources and execute some commands inside containers or manage containers with dockerd, for example:

- reload php-fpm if memory usage is too high
- reload php-fpm if no free space in /dev/shm
- restart container if status is not running
- remove all containers

CHAPTER 1

Table of contents

CHAPTER 2

Quickstart

Using monit-docker in Docker with crond

```
docker-compose up -d
```

See docker-compose.yml and MONIT_DOCKER_CRONS environment variable to configure commands.

CHAPTER 3

Installation

```
pip install monit-docker
```

CHAPTER 4

Environment variables

Variable	Description	Default
`MONIT_DOCKER_CONFIG`	Configuration file contents(e.g. `export MONIT_DOCKER_CONFIG="$(cat monit-docker.yml)"`)	
`MONIT_DOCKER_CONFIGFILE`	Configuration file path	/etc/monit-docker/monit-docker.yml
`MONIT_DOCKER_LOGFILE`	Log file path	/var/log/monit-docker/monit-docker.log
`MONIT_DOCKER_RUNTIMEDIR`	Runtime directory path	/run/monit-docker

CHAPTER 5

Sub-command: monit

5.1 Basic commands

Restart containers with name starts with foo if memory usage percentage > 60% or cpu usage percentage > 90%:

```
monit-docker --name 'foo*' monit --cmd-if 'mem_percent > 60 ? restart'
--cmd-if 'cpu_percent > 90 ? restart'
```

Stop containers with name starts with bar or foo and if cpu usage percentage greater than 60% and less than 70%:

```
monit-docker --name 'bar*' --name 'foo*' monit --cmd-if '60 > cpu_percent < 70
? stop'
```

Kill containers with name starts with bar and status equal to pause or running:

```
monit-docker --name 'bar*' monit --cmd-if 'status in (pause,running) ? kill'
```

You can also use status argument, for example, restart containers with status paused or exited:

```
monit-docker -s paused -s exited monit --cmd 'restart'
```

Generate containers pidfile:

```
monit-docker monit --rsc pid
```

Reload php-fpm in container with image name contains /php-fpm/ if memory usage greater than 100 MiB:

```
monit-docker --image '*/php-fpm/*' monit --cmd-if 'mem_usage > 100 MiB ? (kill
-USR2 1)'
```

Reload php-fpm in container with image name contains /php-fpm/ if /dev/shm percentage usage greater than 80%:

```
monit-docker --image '*/php-fpm/*' monit --cmd '(bash -c "[ $(df /dev/shm |
sed \"s/\%//;\$!d\" | awk \"{print \$5}\") -gt 80 ] && kill -USR2 1")'
```

5.2 Advanced commands with configuration file or environment variable MONIT_DOCKER_CONFIG

5.2.1 Run commands with aliases declared in configuration file (e.g.: monit-docker.yml.example):

Restart container id 4c01db0b339c if condition alias @status_not_running is true:

```
monit-docker --id 4c01db0b339c monit --cmd-if '@status_not_running ? restart'
```

Execute commands alias @start_pause containers with name starts with foo if condition alias @status_not_running is true:

```
monit-docker --name 'foo*' monit --cmd-if '@status_not_running ? @start_pause'
```

Remove force container group php if status is equal to running:

```
monit-docker --ctn-group php monit --cmd-if 'status == running ?
@remove_force'
```

Restart containers group nodejs if memory usage percentage > 10% and cpu usage percentage > 60%:

```
monit-docker --ctn-group nodejs monit --cmd-if '@mem_gt_10pct_and_cpu_gt_60pct
? restart'
```

Remove force all containers:

```
monit-docker monit --cmd '@remove_force'
```

5.3 Container informations with exit codes

5.3.1 Container status

Run command below to get status with exit code for container named foo_php_fpm:

```
monit-docker --name foo_php_fpm monit --rsc status
```

An error occurred if exit code is greater than 100.

Exit code	Description
0	Running
10	Created
20	Paused
30	Restarting
40	Removing
50	Exited
60	Dead
114	Not found

5.3.2 Container CPU usage percentage

Run command below to get CPU usage percentage with exit code for container named foo_php_fpm:

```
monit-docker --name foo_php_fpm monit --rsc cpu_percent
```

An error occurred if exit code is greater than 100.

5.3.3 Container memory usage percentage

Run command below to get memory usage percentage with exit code for container named foo_php_fpm:

```
monit-docker --name foo_php_fpm monit --rsc mem_percent
```

An error occurred if exit code is greater than 100.

5.4 monit-docker with M/Monit

We can also monitoring containers cpu_percent and mem_percent resources with M/Monit.

5.4.1 Configuration examples

```
check program docker.foo_php_fpm.status with path "/usr/bin/monit-docker --name foo_
↪php_fpm monit --rsc status"
    group monit-docker
    if status = 114 for 2 cycles then alert # container not found
    if status != 0 for 2 cycles then exec "/usr/bin/monit-docker --name foo_php_fpm␣
↪monit --cmd restart" # container not running

check program docker.foo_php_fpm.cpu with path "/usr/bin/monit-docker -s running --
↪name foo_php_fpm monit --rsc cpu_percent"
    group monit-docker
    if status > 100 for 2 cycles then alert
    if status > 70 for 2 cycles then alert
    if status > 80 for 4 cycles then exec "/usr/bin/monit-docker --name foo_php_fpm␣
↪monit --cmd reload"

check program docker.foo_php_fpm.mem with path "/usr/bin/monit-docker -s running --
↪name foo_php_fpm monit --rsc mem_percent"
    group monit-docker
    if status > 100 for 2 cycles then alert
    if status > 70 for 2 cycles then alert
    if status > 80 for 4 cycles then exec "/usr/bin/monit-docker --name foo_php_fpm␣
↪monit --cmd '(kill -USR2 1)'"

check program docker.foo_php_fpm.pid with pidfile /run/monit-docker/foo_php_fpm.pid
    group monit-docker
    if changed pid then alert
```

CHAPTER 6

Sub-command: stats

6.1 Basic commands

Get all resources statistics for all containers in json format:

```
monit-docker stats --output json
```


```
{
  "flamboyant_chaplygin": {
    "status": "running",
    "mem_percent": 0.03,
    "net_tx": "0.0 B",
    "cpu_percent": 0,
    "mem_usage": "2.52 MiB",
    "io_read": "3.5 MB",
    "io_write": "0.0 B",
    "net_rx": "25.2 kB",
    "mem_limit": "7.27 GiB",
    "pid": "3943"
  }
}
{
  "practical_proskuriakova": {
    "status": "running",
    "mem_percent": 0.04,
    "net_tx": "0.0 B",
    "cpu_percent": 0,
    "mem_usage": "2.61 MiB",
    "io_read": "24.6 kB",
    "io_write": "0.0 B",
    "net_rx": "25.0 kB",
    "mem_limit": "7.27 GiB",
    "pid": "3990"
  }
}
```

Get all resources statistics for all containers in text format:

```
monit-docker stats --output text
```

```
flamboyant_chaplygin|mem_usage:2.52 MiB|mem_limit:7.27 GiB|mem_percent:0.03|cpu_
↪percent:0.0|io_read:3.5 MB|io_write:0.0 B|net_tx:0.0 B|net_rx:43.5 kB|status:running
practical_proskuriakova|mem_usage:2.61 MiB|mem_limit:7.27 GiB|mem_percent:0.04|cpu_
↪percent:0.0|io_read:24.6 kB|io_write:0.0 B|net_tx:0.0 B|net_rx:43.3␣
↪kB|status:running
```

6.2 Advanced commands with configuration file or environment variable MONIT_DOCKER_CONFIG

Get status and memory usage for group nodejs:

```
monit-docker --ctn-group nodejs stats --rsc status --rsc mem_usage
```