Repairman Documentation

Release 1

Wolnosciowiec Team

Jul 01, 2019

Contents:

Keeps an eye on unhealthy and exited containers. Provides Kubernetes-like functionality to non-Kubernetes environments.

Works with docker, docker-compose, in future can possibly work without an issue with any clusters or even on RKT, LXC and others.

Features:

- Automatic restart of unhealthy containers
- Configurable wait time between container restarts
- Maximum restarts in configured time, after that configured longer wait time
- Removing of duplicated services created with hash-prefixes by docker-compose (ex. after watchtower update)
- Notifications to Slack/Mattermost (with configurable levels: DEBUG, INFO, WARNING)
- Configured default settings via environment variables or console switches
- Each service can override default configuration using Docker Labels
- Lightweight and independent! Provides Kubernetes-like functionality to non-Kubernetes environments
- Can run as a docker container
- Health check endpoint

CHAPTER 1

Quick start

1.1 Using docker container

Official docker container is: *quay.io/riotkit/repairman*

Using docker-compose we can define a clean, easy to read YAML file:

```
version: "2"
services:
    autoheal:
        image: quay.io/riotkit/repairman
        environment:
            NAMESPACE: aarchive
            DEFAULT_SECONDS_BETWEEN_RESTARTS: 15
            DEFAULT_FRAME_SIZE: 450
            DEFAULT_MAX_RESTARTS_IN_FRAME: 3
            DEFAULT_SECONDS_BETWEEN_NEXT_FRAME: 1500
            DEFAULT_MAX_CHECKS_TO_GIVE_UP: 50
            DEFAULT_MAX_HISTORIC_ENTRIES: 50
            DEFAULT_ENABLE_DUPLICATED_SERVICES_REMOVING: "true"
            DEFAULT_ENABLE_AUTO_HEAL: "true"
            TZ: Europe/Warsaw
            DEFAULT_NOTIFY_LEVEL: debug
            DEFAULT_NOTIFY_URL: ""
        restart: always
        mem_limit: 80000000 # 80M, 30M is the average
        volumes:
            - /var/run/docker.sock:/var/run/docker.sock
        labels:
            com.centurylinklabs.watchtower.enable: true
            org.riotkit.repairman.enable_autoheal: false
```

```
# running with all default values
sudo docker run -v /var/run/docker.sock:/var/run/docker.sock wolnosciowiec/
↪repairman:latest
```

(continues on next page)

(continued from previous page)

```

# using environment variables to configure
sudo docker run -e DEFAULT_FRAME_SIZE=450 -v /var/run/docker.sock:/var/run/docker.
↪sock wolnosciowiec/repairman:latest

# using console switches
sudo docker run -e DEFAULT_FRAME_SIZE=450 -v /var/run/docker.sock:/var/run/docker.
↪sock wolnosciowiec/repairman:latest --debug --enable-autoheal
```

1.2 Building and installing a Python package

```
git clone https://github.com/riotkit-org/docker-autoheal
cd docker-autoheal
make install
```

1.3 Building a docker image

```
git clone https://github.com/riotkit-org/docker-autoheal
cd docker-autoheal
make build_image
```

1.4 Installing with Python PIP

```
sudo pip install repairman
repairman --help
```

CHAPTER 2

Configuration

Repairman has two scopes of configuration, internally it's called a policy. Application policy is a default policy for each container, and a Regular Policy is a per-container policy that mixes Application policy + container specific modifications.

Example:

- Application global policy has *time between restarts* equal to 180 and *3 maximum restarts*
- The container can modify some values, ex. will want to have *2 maximum restarts instead of 3 restarts*

2.1 Reference

Parameters			
in shell	as docker env variable	as a docker label	description
–debug	NONE	NONE	Console debugging mode
–interval	CHECK_INTERVAL	NONE	How often in seconds to check all containers
–namespace	NAMESPACE	NONE	Containers prefix (ex. compose env name)
–seconds-between-restarts	DEFAULT_SECONDS_BETWEEN_RESTARTS	org.riotkit.repairman.seconds_between_restarts	Seconds to wait until next try
–frame-size-in-seconds	DEFAULT_FRAME_SIZE	org.riotkit.repairman.frame_size_in_seconds	Frame size (time frame in which max restarts can occur)
–max-restarts-in-frame	DEFAULT_MAX_RESTARTS_IN_FRAME	org.riotkit.repairman.max_restarts_in_frame	Maximum restarts in given time (frame)
–seconds-between-next-frame	DEFAULT_SECONDS_BETWEEN_NEXT_FRAME	org.riotkit.repairman.seconds_between_next_frame	Time between frames (for longer wait)
–max-checks-to-give-up	DEFAULT_MAX_CHECKS_TO_GIVE_UP	org.riotkit.repairman.max_checks_to_give_up	After this number, the service will not be monitored
–max-historic-entries	DEFAULT_MAX_HISTORIC_ENTRIES	org.riotkit.repairman.max_historic_entries	Technically, how many events to remember
–enable-cleaning-duplicated-services	ENABLE_CLEANING_DUPLICATED_SERVICES	org.riotkit.repairman.enable_cleaning_duplicated_services	Remove duplicated services with hash prefix created by compose
–enable-autoheal	DEFAULT_ENABLE_AUTO_HEAL	org.riotkit.repairman.enable_autoheal	Enable healing of unhealthy and exited containers
–http-address	HTTP_ADDRESS	NONE	Web server address ex. 0.0.0.0 or 127.0.0.1
–http-port	HTTP_PORT	NONE	Web server port ex. 80 or 8080
–http-prefix	HTTP_PREFIX	NONE	Web server path prefix ex. /something or /SgbaCaVyewq
–notify-url	DEFAULT_NOTIFY_URL	org.riotkit.repairman.notify_url	Slack/Mattermost notification url
–notify-level	DEFAULT_NOTIFY_LEVEL	org.riotkit.repairman.notify_level	Notify level ex. DEBUG, INFO, WARNING
–db-path	DB_PATH	NONE	Path to sqlite3 database or “:memory:”
NONE	TZ	NONE	Docker container timezone ex. Europe/Warsaw
NONE	DOCKER_HOST	NONE	Docker host address or socket
NONE	DOCKER_TLS_VERIFY	NONE	Verify the host against a CA certificate.
NONE	DOCKER_CERT_PATH	NONE	Path to directory with certificates

2.2 Concept of frames and timing

Frame is a time defined by *–frame-size-in-seconds*, ex. 5 minutes. In this time given service can be restarted only *–max-restarts-in-frame*, if it still fails, then it needs to wait *–seconds-between-next-frame* to next restart try.

2.3 Cleaning up duplicated services

When a *v2tec/watchtower* container is updating a service its starting a container with new image version. After compose up, the container is created twice. The *–enable-cleaning-duplicated-services* resolves this problem by stopping and removing a container with hash prefix.

2.4 Changes between restarts

Repairman uses SQLite3, by default a in-memory database is used - *:memory:*, but it is not a problem to use a persistent database by changing the *–db-path*

2.5 Notifications

Notifications can be sent to Slack/Mattermost. There are three levels of verbosity. **Do not confuse with –debug**

Verbosity levels:

- DEBUG: Each container restart info, maximum restarts limit reached in frame, multiple restart failure info, configuration error
- INFO: Multiple restart failure info, configuration error, maximum restarts limit reached in frame
- WARNING: Configuration error, maximum restarts limit reached in frame

CHAPTER 3

From authors

Project was started as a part of RiotKit initiative, for the needs of grassroot organizations such as:

- Fighting for better working conditions syndicalist (International Workers Association for example)
- Tenants rights organizations
- Various grassroot organizations that are helping people to organize themselves without authority

RiotKit Collective