ClearPath MSP
ClearWatch

SOLUTION BRIEF

from ClearPath MSP

CLEARWATCH SOLUTION BRIEF · KEEPER GATEWAY WITH DOCKER

Keeper Gateway with Docker, from the container to the session.

ClearWatch reads Keeper's own health check and log inside the gateway's Docker container, keeps its image current with one press, and reports who connected, when and from where.

PUBLISHED NOVEMBER 2026 LENGTH 6 PAGES APPLIES TO KEEPER GATEWAY WITH DOCKER

EXECUTIVE SUMMARY

Docker says running. Nobody can connect.

A Keeper Gateway runs as a Docker container on a Linux host and carries your privileged access: RDP, SSH, web and database sessions, password rotation and discovery. Docker's health check for it reports only whether it is connected to Keeper, so it can pass while sessions stall or its processes run out of open files.

The ClearWatch agent on the Docker host reads every container, runs Keeper's own read-only health check inside the gateway and reads its log. A built-in Keeper Gateway monitor judges all of it, a newer image is applied with one press, and the Keeper sessions report shows each gateway's sessions by type, hour and place.

KEY TAKEAWAYS

- **Healthy, not just running:** Keeper's health document, its log and the container's limits, as checks.
- **Found by its image:** the gateway is watched as `keeper/gateway`, so a renamed container stays covered.
- **Updates in one press:** through its own compose file, audited, watched step by step.
- **Checks only read:** no check changes the gateway, and an update runs only when someone presses it.

IN THIS BRIEF

01 **The challenge** 2
What Docker and the log leave to you

02 **How ClearWatch watches it** 2
The agent, the card, the Keeper monitor

03 **Updates, made easy** 4
Flag, press, watch; old images

04 **The Keeper sessions report** 5
Sessions by type, hour and place

05 **Getting started** 6
Four steps to a watched gateway

06 **Good to know** 6
Requirements and limits

WHO THIS IS FOR

MSPs and IT teams running Keeper Gateways in Docker, for themselves or at each client.

AT A GLANCE

READ BY
The agent
On the Docker host, through the engine's own socket.

GATEWAY HEALTH
Keeper's own
Its read-only health check, run inside the gateway.

UPDATES
One press
Compose pull and up for that one service only.

SESSIONS
Per gateway
By type, hour and place, over 24 hours to 30 days.

01 THE CHALLENGE

Running is not the same as working

A Keeper Gateway is a container nobody looks at until privileged access stops working. Docker reports it running, and its health check asks only whether it is connected to Keeper. Sessions that never connect, a process out of open files, an old image, or a gateway busy with nobody connected stay out of sight until a technician cannot get in.

THE QUESTION	DOCKER AND KEEPER'S CONSOLE ALONE	CLEARWATCH
Is the gateway healthy, not just running?	Running, and connected to Keeper	Keeper's **whole health document**: last ping, rotations, discoveries, capacity
Is it running out of something?	Read the process limits by hand	Open files and processes **against its own limits**
Are sessions failing?	Search the container log	Given-up sessions and relay failures **counted as checks**
Did it crash or restart?	`docker ps` and `inspect`	Restarts in the **last 24 hours**, error exits, memory kills
Is there a newer image?	Pull and compare by hand	Checked **daily**; updated in **one press**
Who used it, when, from where?	Keeper's audit report, per tenant	The **Keeper sessions report**, per gateway, beside its processor
A gateway at every client?	One SSH session per host	**Every gateway** on one console, filed per client

Healthy to Docker, broken for everyone else

The Keeper Gateway monitor was built after a gateway in the field passed Docker's health check for more than a day while its sessions stalled in reconnect loops and its processes ran out of open files. The signs were in its log, its open files and a processor busy all day. ClearWatch now judges each of them.

02 HOW CLEARWATCH WATCHES IT

The agent on the Docker host, nothing in the gateway

The ClearWatch agent on the Linux host reads Docker's own socket on every check-in and asks the gateway about itself with Keeper's own read-only command. Nothing is installed in the gateway's container.

LINUX HOST

Keeper Gateway

Keeper's image, `keeper/gateway`, started with Docker Compose. It keeps carrying sessions.

→

SAME HOST

ClearWatch agent

Reads every container, runs `health-check --info` inside the gateway and the read-only Keeper Gateway report every 5 minutes.

→

CLEARWATCH

Checks, updates, report

Judges the checks, raises Needs attention or an alert, offers Update, and draws the sessions report.

Every container

State, health, restarts in 24 hours, processor, memory, open files and log size, against its own limits.

Keeper's own health

Connected to Keeper, last ping, rotation and discovery failures, and whether it can take more work.

Sessions

Each session's start, end, type and where from, read from the gateway's own log.

02 HOW CLEARWATCH WATCHES IT, CONTINUED

The Containers card, and a monitor built for Keeper

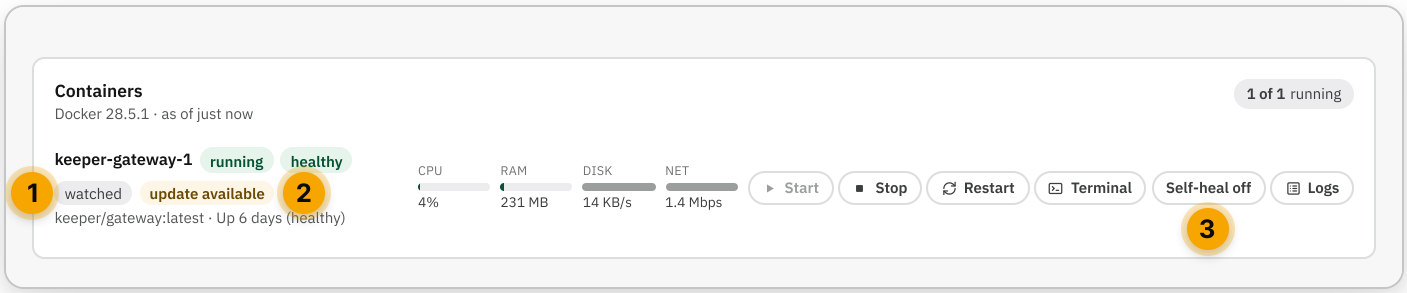


FIGURE 1 The Containers card on the Docker host's Overview tab, with the Keeper Gateway's container.

Every Linux host running Docker gets a Containers card: each container's state, health and use, with Start, Stop, Restart, Terminal and Logs, each asking first. A container running for fifteen minutes is watched automatically 1 : stopped or removed, it fails after two check-ins in a row.

A newer image is flagged on the row 2 ; the name opens its details, updates and image history (section 03). Self-heal is a switch per container 3 . Logs and actions need the Containers permission at control; a shell inside a container is its own level, recorded like a terminal.

The Keeper Gateway monitor

Sixteen checks, one of them the report that collects sessions, finding the gateway by its image so a container Compose renames is still watched. The **Keeper Gateway** template applies it to every host running that image and ships switched off, ready to turn on. Every level is a default you can change for everyone or one device. Self-healing is off on every Keeper check: a restart drops every live session.

CHECK	NEEDS ATTENTION	ALERT
Keeper's own health check health-check --info, run inside the gateway		
Keeper Gateway connected	None	Not connected to Keeper
Last heard from Keeper	120 seconds	300 seconds
Rotations failing, last 10 minutes	25%	50%
Discoveries failing, last 10 minutes	50%	None
Keeper Gateway can take work	Under pressure, or full	None
The container against its own limits		
Open files, the process nearest its limit	50%	80%
Processor	50% or 100% of a core	None: a busy hour is not an outage
Docker health	None	Unhealthy
The gateway's log lines in the last 15 minutes		
"Too many open files"	None	The first line
Sessions given up (circuit breaker, ICE restart timeout)	5 lines	20 lines
Relay (TURN) refresh failures	5 or 20 lines	None
"trickle ICE is disabled", last hour	1 line	None
Sessions from the Keeper Gateway report, every 5 minutes		
Keeper sessions open	50 or 90	None
Keeper tube load	80% or 95%	None
Processor while idle (no session open)	50% of a core	60% of a core

03 UPDATES, MADE EASY

A newer gateway image: see it, press Update, watch

No SSH session and no compose commands typed by hand: it all happens in the container's own window.

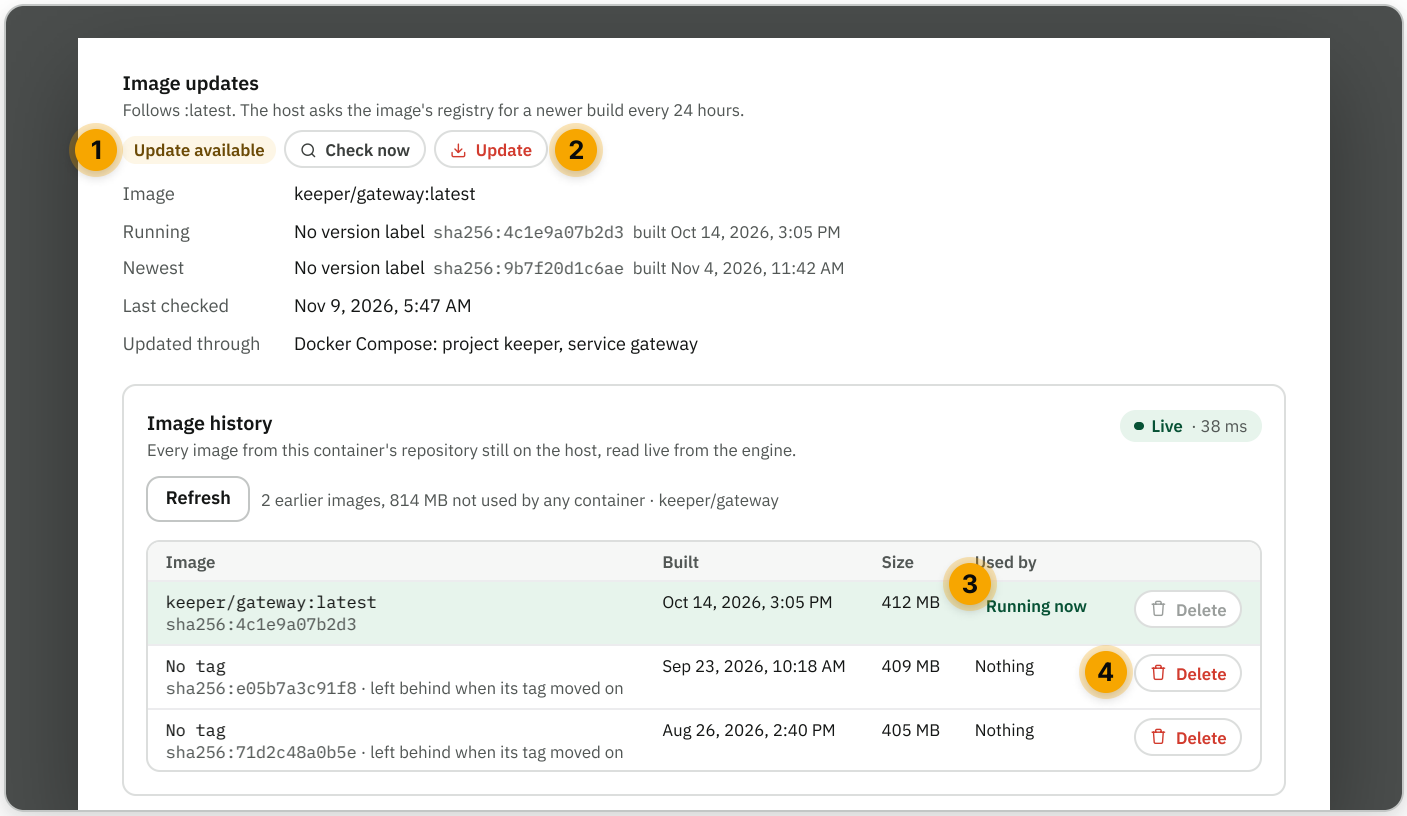


FIGURE 2 The gateway container's window, scrolled to Image updates and Image history.

Once a day the agent asks the image's registry whether a container following `:latest` or `:stable` has a newer build; the image digest decides, never a version label. A newer image is a warning in Needs attention, not a page (1), and Check now asks at once.

Update (2) asks first, then runs `docker compose pull` and `up -d --no-deps` for that one service, from the project's own compose files, so nothing else restarts. It is written to the audit log before it is sent, and the window follows each step (5) with Compose's own output. A step that fails is marked with the host's own words and the steps below it are skipped.

A container update and a patch run are never delivered on top of each other: the second waits until the first settles. Image history keeps the running image (3) and lets you delete the ones left behind, one at a time (4); the host checks again and refuses an image any container uses.

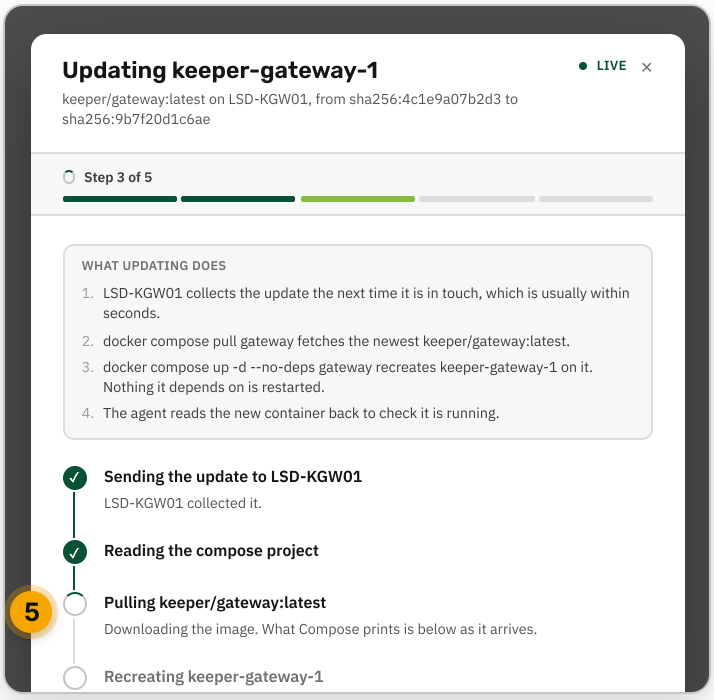


FIGURE 3 The top of the live update window, pulling the new image.

04 THE KEEPER SESSIONS REPORT

What each gateway carried, when, and from where

Keeper sessions is a built-in report under Manage, Reports, over the last 24 hours, 7 days or 30 days in your time zone. Every count is made from the sessions themselves, each type in its own color 1 . Sessions come from each gateway's own log, read every five minutes by the read-only Keeper Gateway report on its host, and are kept thirteen months.



FIGURE 4 The top of the report's first page: sessions started by type over the week, every gateway.

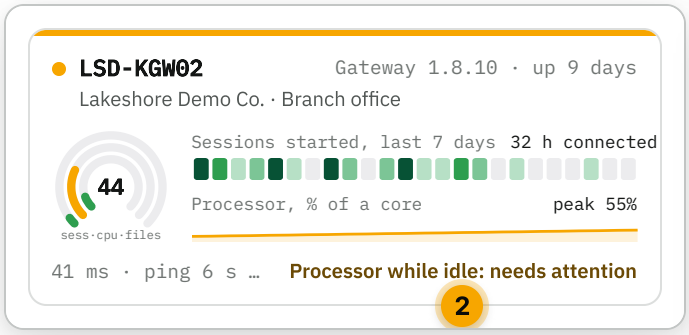


FIGURE 5 A gateway's tile, one per gateway.

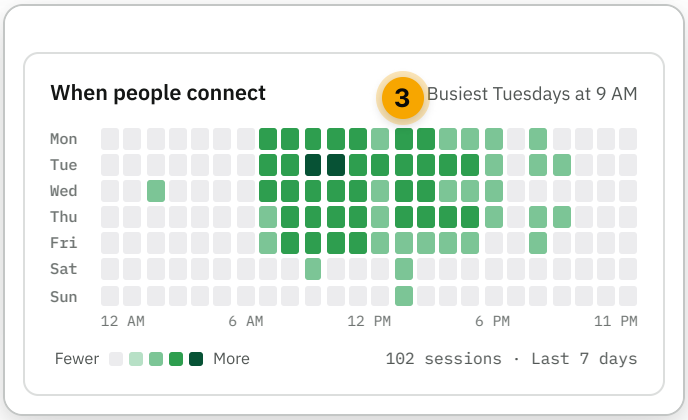


FIGURE 6 Sessions started by day of the week and hour.

Each gateway gets a tile: amber or red appears only where its own check says so 2 . The busiest hour is named 3 . Then each gateway up close, busiest first: a map of where its people connected from, its six figures and its processor against its sessions.

It needs the Reports permission and Containers at See containers, held by whoever operates ClearWatch. Export PDF prints the on-screen preview. The Keeper user is a hash, never an email; Keeper's audit report names people.

05 GETTING STARTED

Four steps to a watched gateway

Nothing is installed inside the gateway and nothing about how it runs has to change. Most of the work is done once, at the host.

1. **Put the ClearWatch agent on the Docker host.** It reads every container on that host through Docker's own socket, the gateway included, on every check-in.
2. **Turn on the Keeper Gateway template.** It finds any container running Keeper's `keeper/gateway` image, so a renamed container is still covered, and applies the Keeper Gateway monitor with its defaults.
3. **Give your team the Containers permission.** Viewing is enough to see the gateway and run the Keeper sessions report; control is what lets someone restart or update it.
4. **Run the gateway from Docker Compose on `:latest` or `:stable`.** That is what lets a newer image be flagged and applied with one press, through the gateway's own compose file.

In summary

Docker can tell you a Keeper Gateway is running. ClearWatch tells you whether it is working: connected to Keeper, carrying sessions, inside its limits and on the newest image, with the sessions it carried in one report. When something needs a hand, the fix is a press in the console rather than an SSH session on the host.

06 GOOD TO KNOW

Requirements

- The ClearWatch agent on the host running the gateway in Docker.
- The Keeper Gateway template on; it finds `keeper/gateway`.
- Update: a Compose container on `:latest` or `:stable`.
- Containers permission: control to update, view for the report.

Limits

- No update for `docker run` or Podman; compose files are never edited.
- Other Keeper containers, such as Keeper Connection Manager, get the general container checks.

See every Keeper Gateway on one console.

Start a free trial, or ask us to turn on the Keeper Gateway monitor with you.

clearwatch.com

ClearWatch™ is a trademark of ClearPath MSP. Keeper and Keeper Security are trademarks of Keeper Security, Inc. Docker is a trademark of Docker, Inc. Screens are rebuilt with demo data.