CUSTOMER GUIDE · DBENK CLOUD COMING SOON

DBenk Worker guide for AutoOps Cloud

How to install, verify, duplicate, update and revoke the DBenk Worker: the agent in your network that runs AutoOps checks without opening an inbound port.

Rp50,000
per server per month, minimum Rp150,000

TCP 443
outbound only, no inbound port

1 command
install on an existing Linux server

Who this guide is for

IT teams installing the DBenk Worker so DBenk AutoOps Cloud can monitor servers in their own network. You need: shell access to one Linux server and the right to add an SSH user on the monitored servers.

Status today

DBenk Cloud and DBenk AutoOps Cloud are **coming soon**: not for sale yet. This guide describes the designed behaviour so your IT team can assess and prepare for it. AVAILABLE means usable today; DESIGNED means it opens together with Cloud.

VERSION
1.0 · 4 November 2026

FOR
IT teams and server admins

LATEST VERSION
lisensi.dbenk.com/panduan

CONTENTS

Table of contents

01 What the DBenk Worker is 3
architecture, who does what, price

02 Requirements 4
host, network, SSH user

03 Installation 5
one command, system service, VM, Docker

04 Checking the Worker is running 6
status, dashboard, connection test

05 Backup Workers and failover 7
adding Workers, automatic takeover

06 Alerts, updates, revoking 8
what they mean and what to do

07 Firewall and network 9
outbound 443 only, no inbound port

08 Troubleshooting 10
symptom, cause, fix

09 Security model 11
keys, signatures, least privilege, audit

All DBenk Cloud guides

- DBenk Worker guide for AutoOps Cloud
- AutoOps Cloud without a Worker: conditions and risks
- DBenk Cloud: what is included

The latest version is always at **lisensi.dbenk.com/panduan**. Prices and the agreement text in this guide come from the same source as the dbenk.com pricing page.

01 OVERVIEW

What the DBenk Worker is

A small agent you install in your own network. It runs AutoOps checks against your servers on behalf of the DBenk cloud, without opening any inbound port.

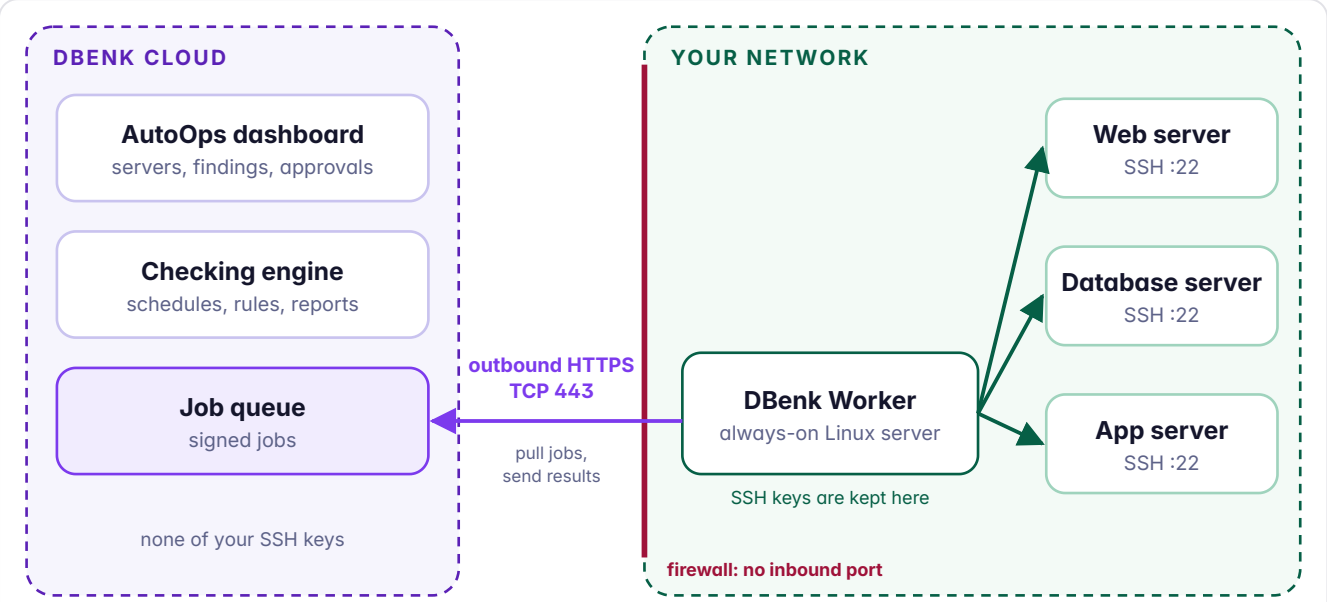


Figure 1. The dashboard and checking engine run in the DBenk cloud. The Worker in your network pulls jobs over outbound HTTPS (443), runs them over SSH against servers on the same network, and sends back the results. SSH keys never leave your network.

Who does what

- **DBenk cloud**: dashboard, check schedules, rules, findings, approvals, history, alerts.
- **DBenk Worker**: keeps the SSH keys, runs jobs signed by the cloud, sends results.
- **Your servers**: no agent of any kind; one minimum-rights SSH user is enough.

Price

DBenk AutoOps Cloud: **Rp50,000** per server per month (Rp500,000 per year), minimum Rp150,000 per month per customer. The DBenk Worker is not billed separately: install as many as you need.

Prices exclude VAT. No free plan; 14-day trial.

02 PREPARATION

Requirements

One always-on server on a network that can reach your servers over SSH.

ITEM	REQUIREMENT	NOTES
Host	An existing always-on Linux server (recommended), or a small VM	No dedicated machine needed. Mini PC/STB follows when ARM64 support for AutoOps opens.
System	Linux x86-64 (amd64)	systemd recommended; without it the Worker uses the user's cron as a watchdog.
Resources	Light: one Go binary, no dependencies	Spare capacity on an existing server is enough; Docker is not required.
Outbound network	HTTPS (TCP 443) to the AutoOps Cloud broker address and `get.dbenk.com`	The broker address is inside the enrolment code. No inbound port.
Internal network	SSH from the Worker to every monitored server	Port 22 or your own SSH port; open to the Worker's address only.
Clock	Synchronised system clock (NTP)	Signed requests are refused when the clock is off by more than 2 minutes.
Account	A normal user (not root) to run the Worker	Root is only needed if you choose a system service.

On the monitored servers

Create one dedicated user, for example `dbenk-autoops`, with no password and no root rights. The default checks only read. Commands that change a server only run for the servers and actions you allow in the dashboard, and every action is logged.

```
# on every monitored server (as an admin)
sudo useradd --create-home --shell /bin/bash dbenk-autoops
sudo passwd --lock dbenk-autoops
sudo install -d -m 700 -o dbenk-autoops -g dbenk-autoops /home/dbenk-autoops/.ssh
# paste the Worker's public key (shown in the dashboard once the Worker is enrolled)
sudo tee -a /home/dbenk-autoops/.ssh/authorized_keys <<< 'ssh-ed25519 AAAA… dbenk-worker'
sudo chown dbenk-autoops: /home/dbenk-autoops/.ssh/authorized_keys
sudo chmod 600 /home/dbenk-autoops/.ssh/authorized_keys
```

Status today

DBenk Cloud and DBenk AutoOps Cloud are **coming soon**: not for sale yet. This guide describes the designed behaviour so your IT team can assess and prepare for it. AVAILABLE means usable today; DESIGNED means it opens together with Cloud.

03 INSTALLATION

Install on a Linux server: one command

The same installer the DBenk Worker uses today: download the signed binary, enrol, start automatically after a reboot.

1 **Create an enrolment code** DESIGNED

In the AutoOps Cloud dashboard open **Workers › Add Worker**. You get a one-time code `DBK-…` valid for 24 hours that carries the broker address. DBenk only stores its hash.

2 **Run the installer as a normal user** AVAILABLE

```
curl -fsSL https://get.dbenk.com/worker | sh -s -- --kode DBK-XXXX-XXXX
```

The installer checks the binary's SHA-256 hash against the signed manifest, creates an ed25519 key pair on this machine (the private key never leaves it), enrols with the broker, and installs autostart: the user's systemd when available, otherwise cron `@reboot` + a 5-minute watchdog.

Want to see the steps first? Add `--dry-run`. Give it a recognisable name with `--nama worker-dc1`.

3 **Optional: system service** AVAILABLE

```
sudo ~/.local/bin/dbenk-worker layanan pasang --sistem --untuk $USER
```

The Worker still runs as that user, but the system's systemd manages it: it starts without anyone logged in.

4 **Add servers in the dashboard** DESIGNED

Enter the server address (internal IP or host name), SSH port and the user `dbenk-autoops`. The dashboard shows the Worker's public key to paste into `authorized_keys` (page 4), then runs a connection test from the Worker.

VM or Docker

VM AVAILABLE

A small Linux VM on your hypervisor is installed exactly like the Linux server above. Pick a VM that is always on and backed up.

Docker DESIGNED

An official Docker image opens together with AutoOps Cloud: a volume for keys and configuration, automatic restart, no published ports. Until then, use a Linux server or VM.

04 VERIFICATION

Checking the Worker is running

On the Worker host

```
dbenk-worker status
```

Shows the version, enrolled profiles, broker address, autostart state and the last successful contact. The log is in `~/.local/state/dbenk-worker/agent.log` (or `journalctl --user -u dbenk-worker` with the user's systemd).

In the dashboard DESIGNED

The **Workers** page shows each Worker as a card: name, state (online, busy, down), version, system, last seen, and the servers it is serving. A Worker counts as online when it was seen in the last 45 seconds.

Connection test to a server

Once the Worker's public key is in place, press **Test connection** on the server card. The Worker opens SSH as `dbenk-autoops` and runs one read-only check. The result is: success, key refused, host unreachable, or host fingerprint changed.

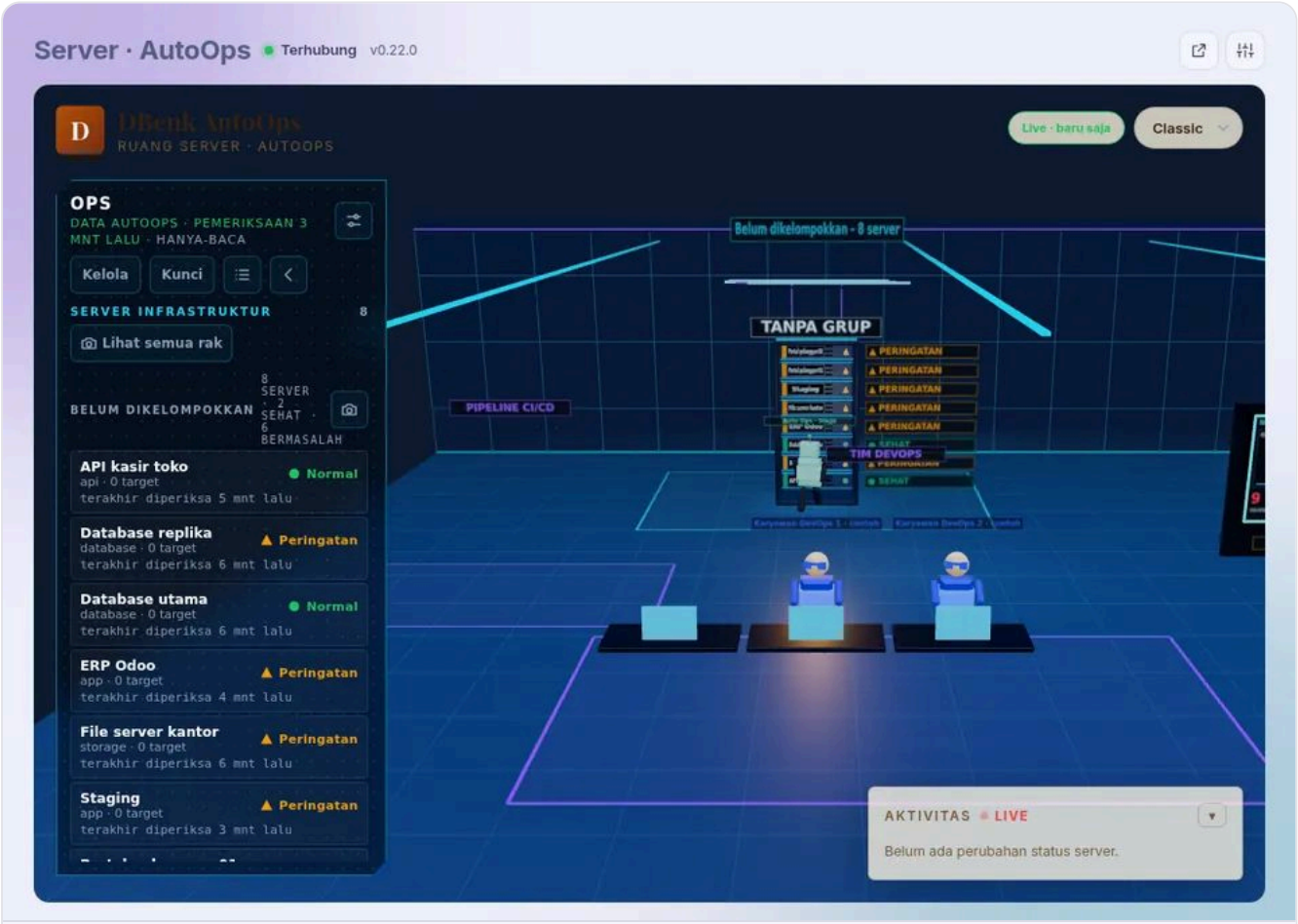


Figure 2. The Server · AutoOps view in DBenk Hub today (sample data from the demo). AutoOps Cloud uses the same server list and health states.

05 BACKUP

A second and third Worker, and how failover works

One Worker is enough to start. To keep monitoring going while one machine is down or being updated, install more Workers on other machines.

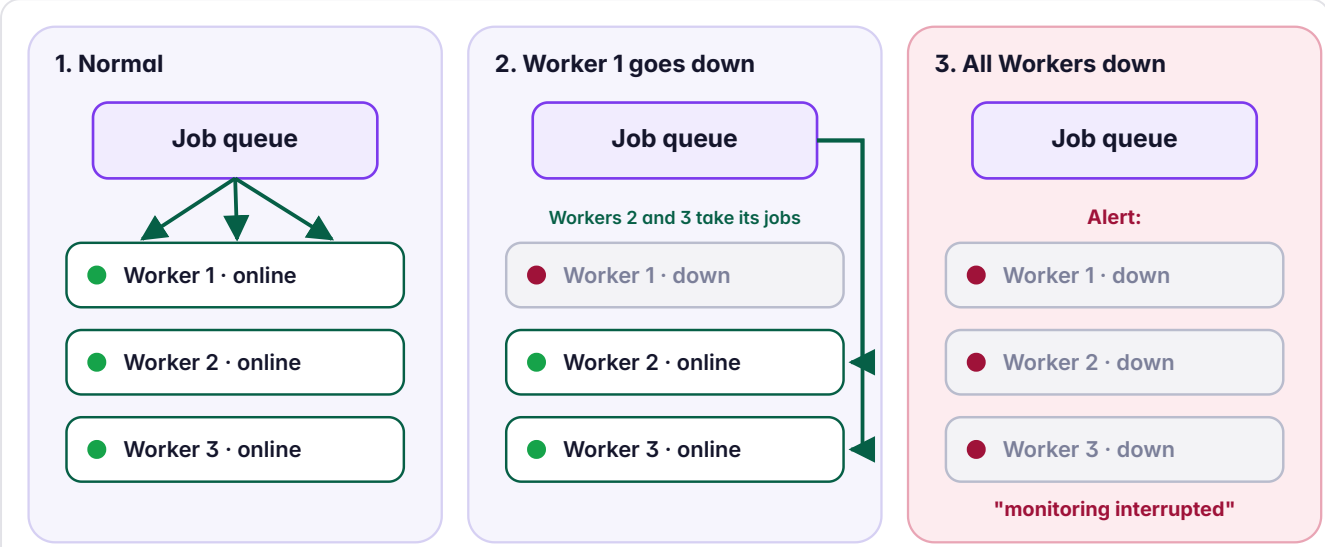


Figure 3. Jobs go to the Workers that are online. When one Worker stops reporting, another one takes over its jobs. When every Worker is down, the dashboard and your notification channels get a *monitoring interrupted* alert.

Adding a Worker DESIGNED

1. Create a new code in **Workers › Add Worker**.
2. Run the same installer command on another machine.
3. Add the new Worker's public key on the monitored servers, or use one shared key per group (a dashboard option).
4. Run a connection test from the new Worker.

Put backups on a different machine, rack or power feed; two Workers on one host do not help when that host goes down.

How failover works DESIGNED

- Every Worker sends a heartbeat to the broker; one not seen for 45 seconds counts as down.
- A job held by a Worker that went down returns to the queue once its lease (60 seconds) runs out, then goes to another Worker; the failed Worker is excluded for that job.
- New jobs only go to online Workers that can reach that server, least loaded first.
- No manual step: a Worker that comes back simply starts taking jobs again.

06 OPERATIONS

Alerts, updates and revoking

Alerts DESIGNED

ALERT	WHEN	WHAT TO DO
Worker down	One Worker is not seen, others are still online	Monitoring continues. Check that machine when convenient.
Monitoring interrupted	No Worker online for the threshold you set	No checks are running. Bring one Worker back (see the troubleshooting table).
Server unreachable	Workers are online but SSH to the server fails	Check the server, the internal firewall and the key.
Outdated version	A Worker cannot update itself	Run the manual update below.

Alerts go to the dashboard and to the channels you choose (e-mail or WhatsApp).

Updates AVAILABLE

The Worker checks the signed release channel `get.dbenk.com/worker` regularly and installs new versions itself: only releases with a valid signature, a newer version (never a downgrade) and a matching hash. Running jobs are not cut off.

```
dbenk-worker update      # update now
dbenk-worker set autoupdate off   # if you
prefer to control it
```

Revoking a Worker AVAILABLE

Press **Revoke** on the Worker card. From then on every request from that Worker is refused (`401 revoked`); the agent stops, removes its autostart and deletes its keys. Also remove its public key from the servers' `authorized_keys`.

```
# remove everything from the machine
curl -fsSL https://get.dbenk.com/worker | s
h -s -- --copot
```

07 NETWORK

Firewall and network

The Worker only needs outbound connections. No port has to be opened from the internet into your network.

DIRECTION	FROM → TO	PORT	PURPOSE
Outbound	Worker → AutoOps Cloud broker	**TCP 443**	pull jobs, send results (HTTPS long-poll)
Outbound	Worker → get.dbenk.com	**TCP 443**	signed updates
Outbound	Worker → NTP server	**UDP 123**	clock sync (or your internal NTP)
Internal	Worker → monitored servers	**TCP 22***	SSH as the user `dbenk-autoops`
Inbound	internet → your network	**—**	**none**

* or your own SSH port. Restrict the servers' SSH port to the Worker addresses only.

Outbound proxy

If outbound internet has to go through a proxy, allow the broker address and `get.dbenk.com` without TLS inspection. The proxy must not rewrite request paths: every Worker request is signed including its path.

Example rule (ufw) on a monitored server

```
sudo ufw allow from 10.0.5.20 to any p
ort 22 proto tcp
# 10.0.5.20 = Worker address; repeat p
er Worker
```

08 TROUBLESHOOTING

Troubleshooting table

SYMPTOM	LIKELY CAUSE	FIX
Installer: code refused	Code already used or older than 24 hours	Create a new code in the dashboard and run again.
Worker shows *down*	Process stopped, machine off, or outbound 443 blocked	`dbenk-worker status`; `dbenk-worker start`; test `curl -I https://get.dbenk.com`.
All requests refused (time)	Machine clock off by more than 2 minutes	`timedatectl set-ntp true` then wait a minute.
Not running after a reboot	User autostart without linger, or cron disabled	`dbenk-worker layanan pasang` or the system service (`--sistem`).
Connection test: key refused	Worker public key missing from `authorized_keys`, or wrong file modes	Paste the key again; `.ssh` 700 and `authorized_keys` 600 owned by `dbenk-autoops`.
Connection test: host fingerprint changed	Server reinstalled, or something sits in the path	Confirm the cause first, then accept the new fingerprint on the server card.
Host unreachable	Internal firewall, routing or wrong SSH port	From the Worker host: `ssh -p 22 dbenk-autoops@<server> true`
`401 revoked` in the log	The Worker was revoked	Enrol again with a new code if it is still needed.
Update failed	Signature or hash mismatch, or a proxy changed the download	The Worker stays on the old version (safe). Check the proxy, then `dbenk-worker update`.
***Monitoring interrupted* alert**	No Worker online	Bring one Worker back; install a second Worker on another machine so it does not happen again.

Still stuck? Send the output of `dbenk-worker status` and the last 50 log lines to support. Logs contain no job contents or secrets.

09 SECURITY

Security model

The principle: secrets stay in your network, every step is signed, and the Worker can only do what it is allowed to do.

Keys stay in your network DESIGNED

The SSH keys for your servers are created and stored on the Worker host (0600 files). The DBenk cloud only sends jobs and receives results; it never holds those keys.

Worker identity AVAILABLE

On enrolment the Worker creates an ed25519 key pair on the device; the private key never leaves it. Every request is signed (±120-second window, single-use nonce), so replayed requests are refused.

Signed jobs AVAILABLE

Every job is signed by the broker, bound to one Worker and has an expiry. The Worker refuses jobs that are invalid, meant for another Worker, or expired. There are no free-form commands: only job types the agent knows.

Signed releases AVAILABLE

Binaries come from a signed channel; DBenk's release keys are pinned inside the binary. Releases are rebuilt reproducibly and must match the manifest byte for byte.

Least privilege

The Worker runs as a normal user, not root. On servers, the user `dbenk-autoops` has no password and no root; default checks only read; changing actions only run for servers and actions you allow, with approval.

Audit

Enrolment, permission changes, revocation and every action on a server are logged with time and actor. Worker and broker logs contain no secrets.

Revoke access at any time

Revoke the Worker in the dashboard (immediate) or remove its public key from the servers. Neither needs help from DBenk.