Installation und erste Aktivierung

Für den Serveradministrator · Neuinstallation auf Linux x86_64 mit Docker Compose. Die Freigabe gilt für diese Installation und alle darin angelegten Benutzer.

1 Vor dem Installationslauf

START-HIER.html öffnen. Archiv und .sha256-Datei liegen in 00_Installation. Prüfsumme vergleichen, Linux-Host, DNS, HTTPS-Zertifikate und Zugangsdaten nach dem Installationshandbuch vorbereiten. Den für diese Installation ausgestellten Medlix-Kaufcode bereithalten.

2 Installation abschließen

Nach configure und install darf ACTIVATION_REQUIRED erscheinen. Jetzt die konfigurierte HTTPS-Adresse von Uptime im Browser öffnen. Die Fachanwendung startet nach erfolgreicher Lizenzprüfung.



3 Mit Internetzugang

Kaufcode eingeben und „Online aktivieren“ wählen. Der Server benötigt HTTPS-Zugang zu lizenz.medlix.de. Die Bestätigung und den anschließenden Start abwarten. Danach mit dem bei der Installation eingerichteten Konto anmelden und MFA einrichten.

Erstaktivierung im abgeschotteten Netz

Der Uptime-Server bleibt im internen Netz. Nur das zweite Gerät benötigt Internetzugang. Für die Aktivierung sind weder Dateiübertragung noch Terminalbefehle erforderlich.

1 Anfrage am Uptime-Server öffnen

Kaufcode eingeben und „Mit QR-Code aktivieren“ wählen. Den angezeigten QR-Code vertraulich behandeln. Er enthält den Kaufcode und die signierte Anfrage dieser Installation.



Abbildung mit fiktiven Daten. Verwenden Sie den QR-Code auf Ihrem Serverbildschirm.

2 Auf dem zweiten Gerät bestätigen

Mit der Kamera scannen, die HTTPS-Adresse lizenz.medlix.de prüfen und auf der Medlix-Seite die Freischaltung bestätigen. Den dort angezeigten gruppierten Freischaltcode vollständig übernehmen.

3 Freigabe lokal übernehmen

Am ursprünglichen Uptime-Bildschirm in „Freischaltcode“ eintragen und „Freischaltung bestätigen“ wählen. Nach dem Start zur Anmeldung wechseln.

Betrieb, Wiederanlauf und Fehlerhilfe

Aktivierung und Benutzerkonten

Die Freigabe gehört zur Serverinstallation. Der Administrator legt Konten und Berechtigungen wie im Rollenhandbuch an. Vor Aktivierung sind die Fachoberfläche, API und Hintergrundverarbeitung gesperrt. Die Integrations-API anschließend bei Bedarf separat einrichten.

Neustart und Sicherung

Eine aktivierte Dauerlizenz funktioniert nach einem Neustart auch ohne Lizenzdienst. Beim Containerwechsel das persistente Schlüsselverzeichnis /keys erhalten. Die vollständige Uptime-Sicherung enthält die Lizenzidentität. Backup verschlüsseln, Zugriff beschränken und den Restore auf einem isolierten Ziel prüfen.

„Bereits belegt“ oder ungültiger Freischaltcode

Prüfen, ob der Kaufcode für diese Uptime-Installation ausgestellt wurde. Bei einem Ersatzserver die vorhandene Sicherung prüfen und die Übernahme mit Medlix abstimmen. Bei ungültiger Offline-Bestätigung den vollständigen Code und die ursprüngliche Installation vergleichen. Kaufcodes, QR-Codes und private Schlüssel nicht in Supportmeldungen senden.

Abnahme und Bestandsanlagen

Nach Aktivierung MFA-Anmeldung, Berechtigungen und einen synthetischen Beispielbericht prüfen. Die Neuinstallation benötigt ein leeres Ziel. Für eine Bestandsumstellung Voraktivierung und Umschaltung mit Medlix planen. Beim Austausch eines einzelnen Containers entsteht Neustartzeit.

Stand dieser Ausgabe: R3-Fachanwendung mit vorgeschalteter Lizenzprüfung. Frühere Angaben zur Image-Gleichheit mit dem früheren Serverstand und zur direkten Erstanmeldung beschreiben die frühere Lieferung. Maßgeblich sind die Release-Identität und die Prüfsummen in diesem Paket.

ANWENDEN. VERSTEHEN. NACHVOLLZIEHEN.

Installations- und Betriebshandbuch

Uptime Control 02.15.1

Detaillierte Abläufe, klare Zuständigkeiten und nachvollziehbare Kontrollen.

Ausgabe 06. November 2026
Linux x86_64 + Docker | Browseranwendung

Keine echten Kundendaten in Abbildungen. Nutzungsrechte richten sich nach der gesonderten Vereinbarung.

Inhalt

01 Diese Ausgabe installieren 3

02 Lieferstand und klare Paketgrenzen 3

03 Ziel und Architektur 3

04 Systemstandard und Dimensionierung 4

05 DNS, Netzwerk und TLS 4

06 Host vorbereiten 4

07 Paket prüfen und entpacken 5

08 DNS, Zertifikatsdateien und Erstkonfiguration vorbereiten 6

09 Kundenkonfiguration 6

10 Installationslauf und Ergebnis 7

11 Erstanmeldung und fachliche Einrichtung 7

12 Dateien, Rechte und Zugangsdaten 8

13 Status, Start, Stopp und Protokolle 8

14 Vollständige Sicherung 9

15 Restore auf leerem Ziel 9

16 Zertifikate und Updates 10

17 Fehlerhilfe und Rückweg 10

18 Abnahme, Übergabe und Beendigung 10

19 Weitere Anwendungsserver autorisieren 11

20 Aktuelle Abnahme und abweichende Konfiguration 12

01 Diese Ausgabe installieren

Installationsdatei: Uptime-Control-02.15.1-linux-amd64.tar.gz im Ordner 00_Installation. Das Archiv enthält alle drei geprüften Linux-amd64-Images, Einrichtungswerkzeuge und die aktuellen Installationshandbücher. Der Kunde benötigt keinen C#-Quellcode und keinen Zugriff auf den bisherigen Server. Es werden keine echten Kundendaten und keine Standard-Anmeldepasswörter mitgeliefert.

Das Paket dient der Neuinstallation in ein leeres Ziel. Nicht über eine bestehende Uptime-Installation laufen lassen. Das Service-/Ticketsystem ist freigegeben; die Integration API bleibt bis zur ausdrücklichen Einrichtung ausgeschaltet. Die zusätzliche Servererweiterung ist davon getrennt.

Vor Beginn: Linux x86_64, lokaler Docker-Daemon mit Compose-Plugin, Python3, OpenSSL, ausreichend Speicher und zwei vorbereitete DNS-Namen mit passendem TLS-Zertifikat. Die konkrete Dimensionierung im folgenden Kapitel ist eine Betriebsplanung, keine garantierte Lastgrenze.

02 Lieferstand und klare Paketgrenzen

Dieses Paket installiert Version 02.15.1 auf einem neuen Linux-x86_64-System mit Docker. Die drei Containerabbilder sind anhand ihrer Image-IDs identisch mit dem am 06.11.2026 um 22:33 UTC erfolgreich aktualisierten Serverstand. Die Konfiguration wird neu für den Kunden erzeugt.

Es handelt sich um eine Neuinstallation ohne vorhandene Kunden- oder Betriebsdaten. Die interne Serversicherung liegt getrennt und wird von restore-new dieses Kundeninstallers nicht verarbeitet. Sie besitzt eine andere Struktur.

Enthalten sind kompilierte Anwendung, Datenbank, Identitätsdienst und Installationshilfen. Kein C#-Anwendungsquellcode und keine Entwicklungsumgebung erforderlich. Notwendige Browserdateien und lesbare Installationsskripte gehören technisch zum Betrieb.

Der Installer ist ein Linux-Tararchiv mit install.sh, keine Windows-EXE. Ein rootberechtigter Linux-Administrator richtet DNS, Zertifikate, Firewall und Sicherungsziel ein.

03 Ziel und Architektur

Vollständige Neuinstallation von Uptime Control 02.15.1 auf einem eigenen Linux-Server mit Docker. Linux-/Docker-Administrationskenntnisse und ein freigegebener Wartungsauftrag sind erforderlich. Dieses Paket ist kein Update- oder Migrationstool für einen vorhandenen Kundenbestand.

Komponenten: unverändertes Uptime-Anwendungsimage; PostgreSQL18.6 mit getrennten Datenbanken uptime und identity; Keycloak 26.8.2 im Produktionsmodus; einmaliger Setup-Helfer für Schema, ersten Administrator und eingeschränkte Laufzeitrechte. Die App läuft ohne Root-Rechte und ohne automatische Migrationen. Keine Beispieldaten werden angelegt.

Datenfluss: Browser zur HTTPS-Anwendung auf443; Anmeldung zum HTTPS-Identitätsdienst auf8443; beide Dienste zur internen Datenbank. PostgreSQL veröffentlicht keinen Host-Port. Das interne Datenbanknetz und ein getrenntes Frontendnetz gehören ausschließlich zum jeweiligen Compose-Projekt.

Der Betreiber verantwortet Host, DNS, Zertifikate, Firewall, Uhrzeit, Konten, Sicherungen und Abnahme. Abweichungen von dieser Einzelserverarchitektur vor Produktivnutzung gesondert prüfen. Die Handbücher begründen keine zusätzliche Wartungs-/Hostingvereinbarung.

04 Systemstandard und Dimensionierung

Zielplattform: Linux x86_64; Bereitstellungsreferenz Ubuntu Server 24.04 LTS mit freigegebenen Sicherheitsupdates. Docker Engine mit Linux-Containern und dem Docker-Compose-Plugin erforderlich. Befehlsform: docker compose. Hostwerkzeuge: Python 3.12 oder neuer, OpenSSL, tar, sha256sum. Ein .NET-SDK oder Anwendungsquellcode ist nicht erforderlich.

Planungsstart:4 vCPU,8 GB RAM,100 GB SSD plus getrennte Sicherungskapazität. Das ist keine zugesicherte Benutzer-/Lastgrenze. Importarchive und Berichte bestimmen den tatsächlichen Speicherbedarf. Für Entpacken und docker load zusätzlich Platz für Archiv und unkomprimierte Images vorsehen.

Die lokale Prüfung verwendete Linux-Container unter Docker Engine29.1.3 und Compose2.40.3. Ein kompletter Ubuntu-Kundenhost wurde hier nicht aufgesetzt. Hostrechte, Härtung, reale Firewall, Neustart und Kapazität bleiben Kundenabnahme.

Vor Beginn festhalten: Server/IP, Ansprechpartner, Wartungsfenster, DNS-Namen, Zertifikatsablauf, Speicher-/Backupziel, tolerierbarer Datenverlust (RPO), Wiederanlaufziel (RTO), Benutzer-/Rollenkonzept. Weitere App-Instanzen erst nach passwortgeschützter Serverfreigabe und gesonderter Infrastrukturabnahme anschließen; siehe Kapitel Servererweiterung.

05 DNS, Netzwerk und TLS

Zwei unterschiedliche DNS-Namen auf dieselbe Server-IP richten, beispielsweise uptime.example.com und login.example.com. Standardports:443 und8443. Beide müssen von den vorgesehenen Arbeitsplätzen erreichbar sein. SSH22 nur aus dem Administrationsnetz. PostgreSQL5432, internes KeycloakHTTP8080 und Managementports nicht veröffentlichen.

DNS/NTP und erforderliche ausgehende Update-/Zertifikatsverbindungen nach Betreiberstandard zulassen. Docker-Portveröffentlichungen an der tatsächlichen Host-/Netzwerkfirewall prüfen; eine gewöhnliche Firewallanzeige allein beweist keine Sperre. Belegte443/8443 oder andere Dienstkonflikte vorab lösen.

Benötigt werden fullchain.pem mit Serverzertifikat/Zwischenkette und privkey.pem als unverschlüsselter, dateirechtlich geschützter PEM-Schlüssel für unbeaufsichtigten Start. Das Zertifikat muss beide DNS-Namen als SAN enthalten. Datenträgerverschlüsselung und restriktiver Root-Zugang schützen zusätzlich.

Öffentliche CA: trust_ca leer lassen. Interne CA: PEM-Vertrauenskette angeben und Vertrauen auf Host/Arbeitsplätzen separat einrichten. Der Installer prüft Schlüssel/Zertifikat, SAN, Kette und mindestens sieben verbleibende Gültigkeitstage. Für die Übergabe ausreichende Restlaufzeit und einen Erneuerungsprozess vorsehen. Zertifikatsfehler niemals durch Abschalten der Prüfung umgehen.

06 Host vorbereiten

Für einen neuen, dedizierten Ubuntu-24.04-amd64-Host ohne bestehende Docker-Installation: die folgenden Befehle nacheinander ausführen. Für andere Distributionen oder einen vorhandenen Containerhost gilt die passende Betreiberanweisung. Repository bereits eingerichtet? Nicht doppelt anlegen.

```
sudo apt-get update
```

```
sudo apt-get install ca-certificates curl python3 openssl
```

```
sudo install -d -m 0755 /etc/apt/keyrings
```

```
sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o /etc/apt/keyrings/docker.asc

sudo chmod 0644 /etc/apt/keyrings/docker.asc

echo 'deb [arch=amd64 signed-by=/etc/apt/keyrings/docker.asc] https://download.docker.com/linux/ubuntu
noble stable' | sudo tee /etc/apt/sources.list.d/docker.list

sudo apt-get update

sudo apt-get install docker-ce docker-ce-cli containerd.io docker-compose-plugin

sudo systemctl enable --now docker

sudo docker version

sudo docker compose version
```

Grundlage: https://docs.docker.com/engine/install/ubuntu/ (abgerufen 06.11.2026). Anschließend Versionen protokollieren.

Docker nach der offiziellen Ubuntu-Anleitung aus dem Herstellerrepository installieren. Keine ungeprüften Fremdskripte als Root ausführen. Die Herstelleranleitung bestimmt aktuelle Repositoryschlüssel/Paketnamen. Erforderlich: Engine/CLI, containerd und Compose-Plugin. Dockerzugang ist weitreichender Hostzugang und gehört ausschließlich in die technische Administration.

Voraussetzungen kontrollieren:

```
uname -m

python3 --version

openssl version

docker version

docker compose version

docker info

df -h

timedatectl status
```

Erwartet: x86_64, funktionierender lokaler Linux-Daemon, ausreichend Speicher und synchronisierte Uhr. Entfernte Docker-Kontexte werden vom Installer verweigert.

Der Installer richtet keinen externen Sicherungsdienst, keine Firewallregel und kein Monitoring automatisch ein. Diese Aufgaben vor Betriebsfreigabe ausdrücklich übernehmen. Ein bereits gemeinsam genutzter Host erfordert gesonderte Konflikt-/Rechteprüfung.

07 Paket prüfen und entpacken

Das Archiv aus 00_Installation samt .sha256-Datei sicher auf den neuen Server kopieren. Prüfsumme über einen unabhängigen vereinbarten Kanal vergleichen. Vorkompilierte Container-Abbilder sind enthalten; docker load benötigt keine Registryverbindung. Host-/Dockerinstallation und Zertifikate können weiterhin Netzzugang benötigen.

Im Übertragungsordner:

```
sha256sum -c Uptime-Control-02.15.1-linux-amd64.tar.gz.sha256

sudo install -d -m 0755 /opt/uptime-control-release
```

```
sudo tar -xzf Uptime-Control-02.15.1-linux-amd64.tar.gz -C /opt/uptime-control-release

cd /opt/uptime-control-release

sudo sh install.sh check
```

Erwartet: PACKAGE_AND_LOCAL_DOCKER_OK. Bei Prüfsummenfehler nicht fortfahren. release.json enthält Einzeldateihashes und exakte Image-IDs; Paket und Manifest müssen aus vertrauenswürdiger Quelle stammen. SHA256 ist keine digitale Herstellersignatur.

Der spätere Betriebsordner /opt/uptime-control muss neu und leer sein. Paket- und Betriebsordner getrennt halten. Das Archiv enthält keine Kundendaten, privaten Zertifikate oder vordefinierten Produktionskennwörter.

08 DNS, Zertifikatsdateien und Erstkonfiguration vorbereiten

1. Zwei DNS-Namen beim Betreiber auf dieselbe Serveradresse zeigen lassen. Auch von Browserarbeitsplätzen müssen beide Namen auflösbar und erreichbar sein. Standard: Anwendung 443, Anmeldung 8443.

2. Ein PEM-Zertifikat einschließlich Zwischenzertifikaten und passendem privaten Schlüssel für beide Namen von der zuständigen CA bereitstellen. Bei interner CA die Vertrauenskette zusätzlich auf Clients und für den Installer bereitstellen. Kein selbstsigniertes Testzertifikat ungeprüft produktiv einsetzen.

3. Dateien geschützt auf dem Host ablegen, beispielsweise /etc/uptime-tls/fullchain.pem und /etc/uptime-tls/privkey.pem. Den privaten Schlüssel ausschließlich der technischen Administration zugänglich machen; der Installer legt seine Laufzeitkopien mit den erforderlichen Rechten an.

4. Docker-Portfreigaben gesondert in der Netzwerkfirewall berücksichtigen; eine einfache UFW-Regel allein kann wegen Docker-Portweiterleitung unzureichend sein. Aus dem erlaubten und aus einem gesperrten Netz testen. Datenbankport 5432 nicht nach außen veröffentlichen.

5. customer.example.json nach customer.json kopieren, mit einem Texteditor bearbeiten und alle elf Felder kontrollieren. Beispiel: sudo cp customer.example.json customer.json; danach sudo nano customer.json.

6. app_host und identity_host enthalten nur Namen, keine https://-Vorsilbe. certificate und private_key sind absolute Linux-Pfade. trust_ca bleibt bei öffentlicher CA leer; sonst den Pfad zur CA-Datei angeben. admin_email muss zur verantwortlichen Person gehören.

7. Nach configure die Bestätigung CONFIGURED abwarten. Das persönliche Erstkennwort wird erst hierbei verdeckt abgefragt. Kein Anbieter-Freigabepasswort als Benutzerkennwort verwenden.

09 Kundenkonfiguration

customer.example.json nach customer.json kopieren. Enthaltene Beispielwerte vollständig ersetzen. Keine Kennwörter in die Datei schreiben.

app_host/identity_host: unterschiedliche DNS-Namen ohne Schema/Pfad. app_port=443 und identity_port=8443: Standardports. bind_address=0.0.0.0: für vorgesehene Arbeitsplätze erreichbar, durch Firewall schützen.127.0.0.1 ist nur für lokale Tests.

admin_username: persönliches/verantwortetes Erstkonto. admin_name/admin_email: tatsächliche benannte Person. certificate/private_key: absolute Hostpfade zu PEM-Dateien. trust_ca: leer bei öffentlicher CA, sonst absolute PEM-Vertrauenskette.

```
sudo sh install.sh configure --directory /opt/uptime-control --config customer.json
```

Erstkennwort verdeckt zweimal eingeben,12 bis128 Zeichen. Individuell erzeugten langen Wert im Passwortmanager verwahren und getrennt übergeben. Technische Kennwörter/OAuth-Geheimnis erzeugt der Installer zufällig; sie werden nicht ausgegeben.

CONFIGURED bestätigt die Vorbereitung, noch keine betriebsbereite Anwendung. state.json enthält technische Geheimnisse im geschützten Betriebsordner. Diese Datei niemals in Supporttickets oder PDFs kopieren.

10 Installationslauf und Ergebnis

Vor Start: Zielordner dieser neuen Installation zugeordnet, DNS/Zertifikate korrekt, Ports frei, Root-Auftrag und Speicher ausreichend.

```
sudo sh install.sh install --directory /opt/uptime-control
```

Ablauf: exakte Images laden; neue PostgreSQL-Instanz und zwei leere Datenbanken/Rollen; produktiven Keycloak starten; neuen Realm importieren; Passwort-/OTP-Nachweise konfigurieren; Uptime-Schema migrieren; ersten Administrator anlegen; eingeschränkte App-Rechte vergeben; App starten; zertifikatsgeprüfte HTTPS-Bereitschaft kontrollieren.

Erfolg: UPTIME_READY und INSTALLATION_READY. installed.json dokumentiert nur technischen Start. Nach Erfolg wird das Erstkennwort aus der Realm-Importdatei entfernt; technische Geheimnisse bleiben geschützt erhalten. Wiederholtes install bei fertiger Installation wird verweigert. Normaler Start erfolgt über start.

Bei Abbruch keine unbekannten Ordner/Volumes löschen. Status und Logs prüfen. Kein allgemeiner automatischer Rollback ist enthalten. Vor installed.json kann ein erneuter Lauf den begonnenen Aufbau fortsetzen; zuerst Ursache und vorhandenen Zustand kontrollieren. Vollständiger Neuversuch nur an eindeutig zugeordneter isolierter Installation nach Sicherungsentscheidung.

11 Erstanmeldung und fachliche Einrichtung

Konfigurierte HTTPS-Adresse öffnen und Weiterleitung zum getrennten Anmeldedienst prüfen. Mit Erstkonto anmelden, OTP in persönlicher Authenticator-App einrichten, bestätigen und erneut anmelden. QR-Geheimnis nicht fotografieren/versenden. Bei Codefehler Uhrzeit synchronisieren und nächsten Code abwarten.

Prüfen: richtige Person, Administration verfügbar, Demo-Modus aus, Service- und Ticketmodul freigegeben, separate Servererweiterung nicht automatisch freigegeben, keine Beispiel-Flughäfen/Anlagen. Ein synthetisches Testkonto über die Anwendung anlegen und tatsächliche Identitätsanlage prüfen. uptime.provisioner ist ein technisches Konto ohne fachliche Arbeitsrolle; nicht deaktivieren.

Fachliche Reihenfolge: Berichtsabsender/Unternehmen; Flughäfen mit Land/IANA-Zeitzone; Bereiche; Anlagen mit Codes/Seriennummern; Vertragsversionen mit Gültigkeit, Betriebsfenstern, Kulanz und Zielwerten; persönliche Benutzer und passende Zuweisungen.

Mit freigegebenen synthetischen Dateien Vorschau/Import, berechtigte Servicekorrektur, Bericht, PDF/CSV und verweigerte Prüferänderung testen. Testdaten nicht als produktive Leistungsdaten belassen. Erst nach dokumentierter Freigabe echte Dateien importieren.

12 Dateien, Rechte und Zugangsdaten

state.json: Parameter, Projektkennung und technische Geheimnisse. compose.json: abgeleitete Containerkonfiguration mit Geheimnissen. tls/: Zertifikate/privater Schlüssel. keys/: Data-Protection-Schlüssel und gegebenenfalls geschützte Aktivierungsnachweise. init.sql: DB-Rollen. realm.json: Realm-/Clientkonfiguration. setup/: kompilierter Initialisierungs-/Prüfhelfer.

Das eindeutig benannte PostgreSQL-Volume enthält uptime und identity. Ein Volume ist keine Sicherung. release.json dokumentiert Image-IDs und Paketstand. Betriebsdateien nicht mit allgemeinen Kundenunterlagen versenden.

Auf Linux: oberstes Betriebsverzeichnis0700; keys gehörtUID/GID1000; TLS-Verzeichnis erlaubt Containergruppe1000 Lesen. Nach Umzug Besitz/Modus prüfen; kein chmod777. Docker-/Rootzugang streng begrenzen.

Getrennt sicher übergeben: Linux/SSH, initialer App-Administrator, Keycloak-Notfalladministrator, Backup-/Passwortmanagerzugang. Im Abnahmeformular nur Kanal/Empfang vermerken, keine Werte. Technische Passwörter nicht allein in state.json ändern: Datenbank/Identität und Laufzeitkonfiguration müssen zusammenpassen.

13 Status, Start, Stopp und Protokolle

Im erhaltenen Paketordner ausführen:

```
sudo sh install.sh status --directory /opt/uptime-control
```

```
sudo sh install.sh stop --directory /opt/uptime-control
```

```
sudo sh install.sh start --directory /opt/uptime-control
```

status zeigt nur dieses Projekt und prüft HTTPS-Bereitschaft. stop beendet App, Identität und DB kontrolliert ohne Volumes zu löschen. start startet diese Dienste. Neustart im Wartungsfenster über stop/start.

Für gezielte Logs benötigen docker compose-Befehle die Projektkennung aus state.json und die konkrete compose.json. Keine vollständige aufgelöste Konfiguration in Tickets kopieren. Logs rotieren je Dienst mit10MB/fünf Dateien; dies ersetzt keine zentrale Langzeitprotokollierung.

Restart-Richtlinie unless-stopped: Docker beim Hostboot aktivieren und realen Neustart kundenseitig testen. Absichtlich gestoppte Dienste bleiben entsprechend Docker-Semantik gestoppt. Überwachen: Bereitschaft, Anmeldung, Speicher, Zertifikatsablauf, Sicherungserfolg. Schwellenwerte/Alarmempfänger festlegen.

Installations-, Sicherungs- und Wiederherstellungsbefehle nicht parallel ausführen. Ein verantwortlicher Operator koordiniert das Wartungsfenster.

14 Vollständige Sicherung

Das Werkzeug sichert beide Datenbanken, Anwendungsschlüssel, TLS-Material und erforderliche Konfiguration. Ein Download der Anwendungssicherung aus der Oberfläche ersetzt dieses vollständige Betriebsbackup nicht.

Ziel außerhalb des Betriebsordners anlegen. Backup pausiert App und Anmeldung; Wartungsfenster vorsehen. PostgreSQL bleibt für konsistente Dumps verfügbar. Nach Erfolg/Fehler wird der Wiederstart versucht. Die fertige Archivbezeichnung entsteht erst nach erfolgreicher Erstellung aller Teile.

```
sudo install -d -m 0700 /var/backups/uptime-control
```

```
sudo sh install.sh backup --directory /opt/uptime-control --backup-dir /var/backups/uptime-control
```

Erwartet: BACKUP_COMPLETE mit Pfad und Bereitschaft. Zu jeder.tar.gz entsteht eine.sha256-Datei. Das Archiv enthält höchst sensible Schlüssel, Kennwörter und Identitäten. Es ist NICHT automatisch verschlüsselt. Vor externer Speicherung mit dem vereinbarten Backupverfahren verschlüsseln; Schlüssel getrennt verwahren.

Eine Kopie auf demselben Server schützt nicht vor Serververlust. Externe Kopie, Integritätsprüfung, Aufbewahrung, Restoreprobe und Alarmierung dokumentieren. Kein Timer/Cloud-/NAS-Zugang wird automatisch eingerichtet. Nach erster Sicherung und relevanten Änderungen Restore testen; Rotation nur mit geprüftem neuen Sicherungsstand.

15 Restore auf leerem Ziel

Benötigt: passendes Paket samt Images, vollständiges Backup/Prüfsumme, TLS/CA/DNS-Zugang, freier Speicher und neuer leerer Betriebsordner. restore-new überschreibt keinen Bestand. Version muss zum Backup passen.

Vor Umschaltung Altinstanz stoppen und Rückweg sichern. Gleiche Ports auf demselben Host können nicht gleichzeitig belegt sein. Auf Ersatzhardware DNS erst nach erfolgreicher Abnahme umstellen; niemals versehentlich zwei beschreibbare Instanzen anbieten.

```
sha256sum -c uptime-backup-ZEITSTEMPEL.tar.gz.sha256
```

```
sudo sh install.sh restore-new --directory /opt/uptime-control-restored --archive
/sicherer/pfad/uptime-backup-ZEITSTEMPEL.tar.gz
```

Beispielpfad/ZEITSTEMPEL ersetzen. Werkzeug prüft interne Dateihashes, erzeugt neues Compose-Projekt/DB, restauriert beide Datenbanken und Schlüssel/Konfiguration und startet Dienste. RESTORE_READY bedeutet technische Bereitschaft.

Danach mit bestehendem Konto und demselben Authenticator anmelden; Stammdaten, Rollen, Importarchive, Berichtsrevisionen, Exporte und Modulsperren prüfen. Datenmengen/fachliche Stichproben mit Sicherungsstand vergleichen. Erst danach Benutzer freigeben. Bei Fehlern Ziel isolieren, Altinstallation/Backup erhalten und keine Ersatz-Leerdatenbank als Erfolg ausgeben.

16 Zertifikate und Updates

Ablauf frühzeitig überwachen. Ersatzzertifikat auf beide Namen, Kette, passenden Schlüssel und Laufzeit prüfen. Vollständige Sicherung erstellen. Im Wartungsfenster Dienste stoppen, TLS-Dateien unter Erhalt von Rechten/Besitz ersetzen, wieder starten; HTTPS und Anmeldung von regulärem Client ohne Zertifikatsausnahme testen.

Domain-/Port-/CA-Wechsel betrifft OAuth-Weiterleitungen, Keycloak-Hostname, Vertrauen und Appkonfiguration gemeinsam. Erfordert gesondert geprüften Änderungsplan; keine automatische Migration durch bloße Dateieditierung.

Dieses Paket enthält keinen allgemeinen Bestandsupdater. Spätere Updates nur mit definiertem Ausgangsstand, geprüftem Paket, Sicherung, Migrationstest, Rückweg und Abnahme. Images nicht auf latest setzen und keine Einzelkomponente unkontrolliert austauschen.

OS-/Docker-Sicherheitsupdates durch Betreiber pflegen; danach Bereitschaft, Login, Rollen, Import/Export und Backup prüfen. Vertragsregeländerungen separat versionieren. Kundenänderungen am Quellcode verlassen den geprüften Lieferstand.

17 Fehlerhilfe und Rückweg

Prüfsummenfehler: Transfer/Referenz prüfen, Manifest nicht passend umschreiben. Remote Docker: bewusst lokalen Kontext wählen. Ziel nicht leer: neuen Ordner verwenden oder Bestand behandeln, nicht pauschal löschen.

Identität nicht bereit: DB, Speicher, TLS-Rechte, Hostnamen und Logs prüfen. Bereitschaft/TLS fehlerhaft: Kette, SAN, CA-Vertrauen und App-/DB-Verbindung prüfen; keine Sicherheitsprüfung abschalten.

Login ohne Zugriff: Benutzer und aktive Bereichszuweisung prüfen. OTP abgelehnt: Zeit synchronisieren/neuen Code abwarten. Benutzeranlage fehlerhaft: technisches Provisioningkonto, vollständiges Realm-Profil und HTTPS-Verbindung prüfen. Service-/Ticketmodul unerwartet gesperrt: Lieferkonfiguration und Rollen prüfen. Nur die zusätzliche Servererweiterung benötigt ihre gesonderte Freigabe.

Backupfehler: Status/Platz prüfen und vollständige Sicherung wiederholen. Restorefehler: unveränderte Altinstanz/Backup erhalten. Keine pauschale manuelle SQL-Korrektur fachlicher Daten. Supportbeleg: Zeit/Zeitzone, Version, Phase, bereinigte Meldung/Referenz. Keine state.json, privaten Schlüssel oder Dumps ungeschützt versenden.

18 Abnahme, Übergabe und Beendigung

Separates Abnahmeformular ausfüllen: gültiges TLS, URLs, externe DB-Sperre, OTP, Benutzer/Rollen, Stammdaten/Regeln, Import/Korrektur/Export, Modulsperren, vollständiges Backup mit verschlüsselter externer Kopie, Restore samt Identitäten/Schlüsseln, Hostreboot und Betriebsverantwortliche.

Paket-Prüfsumme, Image-IDs, tatsächliche nicht geheime Systemkonfiguration, Abnahmedatum und Ansprechpartner aufbewahren. Kennwörter/Recoverymaterial getrennt im Tresor. RPO/RTO/Kapazität erst nach ausdrücklicher Vereinbarung und Prüfung als zugesichert behandeln.

Außerbetriebnahme: Aufbewahrung/Export abstimmen, vollständiges Backup prüfen, Zugriff kontrolliert sperren. stop löscht nichts. Datenvolumes/Identitäten/Schlüssel löschen ist gesonderter irreversibler Vorgang außerhalb der Installation.

Quellcode nur bei entsprechender Vereinbarung übergeben. Für den Binärbetrieb ist er nicht erforderlich. Fremde Lizenzhinweise und Komponentenübersicht erhalten.

19 Weitere Anwendungsserver autorisieren

Die Servererweiterung verwendet ihr eigenes Anbieterpasswort. Die Erstinstallation erhält eine neue Installationskennung und ein leeres Serverregister. Es wird kein zusätzlicher Server automatisch gestartet. Die bestehende Hauptanwendung darf im Modus Single bleiben.

Im entpackten Paketordner auf dem Linux-Hauptserver:

```
sudo python3 modules.py status --directory /opt/uptime-control
```

Eine individuelle Freigabe erzeugen; das Passwort wird verdeckt abgefragt:

```
sudo python3 modules.py server-release --directory /opt/uptime-control --node node2 --output
/sicher/node2.zip
```

Der Ausgabeordner muss geschützt existieren, die Zieldatei darf nicht existieren. Namen: 1 bis 48 Buchstaben, Ziffern oder Bindestriche; node1 ist reserviert. Passwortfehler, doppelte Namen, parallele Freigaben und vorhandene Exportdateien werden abgewiesen. Der lokale Hauptcontainer muss mit dem passenden Image laufen.

Falls die Freigabe gespeichert, der Export aber unterbrochen wurde:

```
sudo python3 modules.py server-export --directory /opt/uptime-control --node node2 --output
/sicher/node2.zip
```

Das vertrauliche ZIP enthält individuelles Permit, Register, Image-Identität und eine Administrationsvorlage. Es enthält kein owner.json und kein Freigabepasswort. Es ist kein fertiger Zweitserver-Installer. Für weitere Server jeweils eigene Namen und Freigaben verwenden.

Zusätzlichen Host anschließen

Vorher private Netzverbindungen, gemeinsame vorhandene PostgreSQL-Datenbank und Keycloak, dauerhaft gemeinsam nutzbare Schlüsselablage, passende TLS-Zertifikate, Lastverteilung und Verbindungsbudgets einrichten. Keine zweite leere Datenbank oder neue Kontenbasis erzeugen.

Das identische Anwendungsimage aus dem Installationsarchiv laden; INSTALLATION.json nennt die genaue Image-ID. compose.additional.yml und eine geschützte runtime.env mit den bestehenden App-/Identitäts-/Datenbankeinstellungen verwenden. Individuelle Freigabe und Register nur lesend einbinden. owner.json bleibt auf dem Haupt-Verwaltungshost. UID 1000 benötigt Zugriff auf gemeinsame Schlüssel und Zertifikate; der zusätzliche Host muss den HTTPS-Identitätsdienst korrekt auflösen und dessen Zertifikat prüfen.

Bei aktivem Ticketmodul denselben geschützten Freischaltnachweis und Schlüsselbestand gemeinsam nutzen. Eine einmalige Schlüsselkopie genügt wegen Rotation nicht. Nach Änderungen das aktuelle Register kontrolliert an weitere Knoten verteilen.

Erst Anmeldung/Sitzungswechsel, Rollen, Import, Speicherung, Berichte, Tickets sowie Knotenausfall und Wiederanlauf auf der realen Infrastruktur abnehmen. Die lokale Prüfung dieser Ausgabe ersetzt keinen physischen Mehrhosttest oder eine Kapazitätszusage.

20 Aktuelle Abnahme und abweichende Konfiguration

Für 02.15.1 wurden Neuinstallation, echter Browserlogin mit MFA, Benutzeranlage, vollständige Sicherung/Wiederherstellung mit 86 identischen Anwendungstabellen und erneute Anmeldung geprüft. Die API wurde zusätzlich mit echten Keycloak-Tokens in 17 Prüfungen abgenommen. Die Kundenumgebung wurde damit nicht vorweggenommen.

Wichtig: Individuelle Compose-/API-Konfiguration separat geschützt sichern und ihren Restore prüfen. restore-new erzeugt die Standardkonfiguration aus dem gesicherten Installationszustand; beliebige nachträglich von Hand geänderte Compose-Werte werden nicht als vollständig wiederherstellbar zugesichert. Die Testabnahme des Standardbackups erfolgte mit deaktivierter API.

Bei öffentlicher Erreichbarkeit gültige Zertifikate verwenden. In einem VPN müssen beide Namen einschließlich Anmeldedienst auflösbar und erreichbar sein. Docker-Portfreigaben separat von Host-Firewallannahmen prüfen. Keine Datenbankports öffnen.

Offizielle Host-Vorbereitung: https://docs.docker.com/engine/install/ubuntu/ und https://docs.docker.com/compose/install/linux/. Stand der Quellenprüfung: 06.11.2026. Im vorhandenen Kundensystem keine fremden Container oder Docker-Pakete ungeprüft entfernen.