Utrzymanie

Kopia, aktualizacja, dostęp zdalny i częste problemy.

- Kopia, aktualizacja i dostęp zdalny: Home Assistant bez otwartego portu 8123
- Home Assistant: częste problemy z urządzeniami, bazą i aktualizacjami

Kopia, aktualizacja i dostęp zdalny: Home Assistant bez otwartego portu 8123

Poziom: średnio

Dotyczy: Home Assistant 2026.9 (Home Assistant OS 18.3 i Container), bkp01 (Rocky Linux 9, Samba), nginx na docker01, FortiGate fw01

Czas: ~90 min

Pokazujemy, jak Home Assistant na `ha01` zabezpieczamy trzema rzeczami: kopią zapasową trzymaną poza maszyną, spokojnym rytmem aktualizacji i dostępem zdalnym wyłącznie przez VPN, bez wystawiania portu do internetu. Wracamy tu przy pierwszej konfiguracji kopii, przed każdą większą aktualizacją i przy pytaniu, jak otworzyć panel z telefonu spoza biura. Konta i uwierzytelnianie dwuskładnikowe opisujemy na końcu.

Skróty użyte na tej stronie (19)

Skrót	Co znaczy
CIFS	Common Internet File System: starsza nazwa protokołu współdzielenia plików Windows, dziś częściej nazywanego SMB, pojawia się w wynikach poleceń FreeIPA dotyczących zaufania z Active Directory
ESPHome	Oprogramowanie, które zamienia ESP32 w gotowy czujnik sieciowy bez pisania kodu od zera, a odczyty udostępnia przez sieć
HTTP	Hypertext Transfer Protocol: protokół stron WWW
HTTPS	HTTP zabezpieczony szyfrowaniem TLS
IKEv2	Internet Key Exchange w wersji 2: nowsza odmiana IKE, wbudowana w klienta VPN Windows, szybciej wraca do życia po zerwaniu łącza niż starsze protokoły VPN
IPsec	Internet Protocol Security: zestaw protokołów szyfrujących i uwierzytelniających ruch sieciowy, na nim stoi tunel VPN typu IKEv2
NAS	Network Attached Storage: dysk sieciowy
NFS	Network File System: udostępnianie katalogów po sieci w świecie Linuksa

Skrót	Co znaczy
NVMe	Non-Volatile Memory Express: protokół komunikacji z dyskami SSD podłączonymi przez złącze PCIe, znacznie szybszy niż starsze SATA i SAS, typowy dla szybkich dysków systemowych
OS	Operating System: system operacyjny
Samba	program dla Linuksa, który udostępnia katalogi w sieci protokołem SMB, tak jak serwer plików Windows
SELinux	Security-Enhanced Linux: dodatkowa warstwa uprawnień w jądrze RHEL
SMB	Server Message Block: protokół udostępniania plików i drukarek w sieci Windows, na nim stoi zwykły udział sieciowy
TLS	Transport Layer Security: szyfrowanie połączeń, np. w HTTPS
TOTP	Time based One Time Password: jednorazowy kod zmieniający się co chwilę, generowany przez aplikację uwierzytelniającą jako jedna z metod dwuetapowego logowania
USB	Universal Serial Bus: standard portów i urządzeń podłączanych do komputera, tu pendrive'y i inne nośniki wymienne
VPN	Virtual Private Network: szyfrowany tunel do sieci firmowej
WebSocket	stałe, dwukierunkowe połączenie przeglądarki z serwerem, przez które Home Assistant na żywo odświeża panele; serwer pośredniczący musi przekazywać nagłówki Upgrade i Connection
YAML	YAML Ain't Markup Language: format plików konfiguracyjnych oparty na wcięciach

Fiszka

□□ KLIKAMY W INTERFEJSIE

Kopia ręczna z interfejsu: Settings, System, Backups

```
Settings → System → Backups → Create backup
```

Kopia jest szyfrowana i zapisywana jako archiwum `.tar`, a do jej odtworzenia potrzebny jest klucz z emergency kit [1]. Klucz pobieramy od razu po pierwszym włączeniu kopii i zapisujemy w Vaultwardenie na `hasla.zielona.example`, poza `ha01` i poza `bkp01` [1]. Kopia bez klucza jest dla nas bezużyteczna, więc klucza nie trzymamy w tym samym miejscu co kopie.

□□ ZRZUT EKRANU DO UZUPEŁNIENIA

Zrzut do uzupełnienia: ekran Settings → System → Backups w `ha01` z oknem tworzenia kopii, widocznym szyfrowaniem i przyciskiem pobrania emergency kit (klucz zasłonięty).

ZMIENIA SYSTEM

Przygotowanie udziału na bkp01: useradd i smbpasswd

```
sudo dnf install --assumeyes samba policycoreutils-python-utils
sudo mkdir -p /backup/ha
sudo useradd --system --no-create-home --shell /usr/sbin/nologin hakopie
sudo smbpasswd -a hakopie
sudo chown hakopie:hakopie /backup/ha
sudo chmod 750 /backup/ha
```

Pakiet `samba` instalujemy tylko wtedy, gdy `bkp01` jeszcze go nie ma. Konto `hakopie` istnieje tylko po to, żeby Home Assistant zapisywał kopie, i nie ma powłoki. Hasło do Samby wpisujemy po pytaniu `smbpasswd` i zapisujemy w Vaultwardenie. Katalog `/backup` jest tym samym, w którym `bkp01` trzyma kopie z `docker01`, jak na stronie Kopia zapasowa aplikacji dockerowych.

dnf = Dandified YUM: menedżer pakietów RHEL/Rocky · **mkdir** = make directory: utwórz katalog · **useradd** = user add: tworzy nowe konto systemowe w Linuksie · **smbpasswd** = SMB password: ustawia hasło konta w Sambie, osobne od hasła systemowego, np. smbpasswd -a NazwaKonta · **chown** = change owner: zmień właściciela · **chmod** = change mode: zmień prawa dostępu

PLIK KONFIGURACYJNY

Definicja udziału Samby: /etc/samba/smb.conf

```
[ha-kopie]
   path = /backup/ha
   valid users = hakopie
   read only = no
   create mask = 0640
   directory mask = 0750
```

Blok dopisujemy na końcu `/etc/samba/smb.conf`, a w sekcji `[global]` ustawiamy grupę roboczą `ZIELONA` i `server min protocol = SMB2`. Cały plik z tą sekcją przeszedł `testparm` w Sambie 4.19.

□□ **read** = bash: wczytaj wiersz z klawiatury do zmiennej, a z opcją -s bez wypisywania wpisanego tekstu na ekran, na przykład dla tokenu API

□□ ZMIENIA SYSTEM

Uprawnienia SELinux i zapora na bkp01: semanage, restorecon, firewall-cmd

```
sudo semanage fcontext --add --type samba_share_t "/backup/ha(/.*)?"
sudo restorecon -Rv /backup/ha
sudo firewall-cmd --add-service=samba --permanent
sudo firewall-cmd --reload
sudo systemctl enable --now smb
```

Etykieta `samba_share_t` oznacza zawartość udostępnianą przez Sambę, a bez odpowiedniej etykiety SELinux może blokować zapis do katalogu [2]. Narzędzie `semanage` pochodzi z pakietu `policycoreutils-python-utils`, który instalujemy razem z Sambą. Usługę Samby i przeglądanie udziałów z `ha01` ograniczamy zaporą `fw01` do sieci serwerów.

□□ **SELinux** = Security-Enhanced Linux: dodatkowa warstwa uprawnień w jądrze RHEL · **semanage** = SELinux manage: zarządzanie regułami SELinux · **restorecon** = restore context: przywróć etykiety SELinux · **firewall-cmd** = sterowanie zaporą firewalld · **systemctl** = system control: sterowanie usługami systemd

□□ KLIKAMY W INTERFEJSIE

Dodanie udziału jako miejsca na kopie: Settings, System, Storage

```
Settings → System → Storage → Add network storage
```

Wybieramy użycie Backup, protokół SMB, serwer 10.20.10.61, nazwę udziału `ha-kopie` oraz konto `hakopie` z hasłem z Vaultwardena [3]. Home Assistant OS obsługuje magazyny NFS i CIFS (SMB) [3]. Nazwę magazynu zostawiamy `bkp01`, żeby w liście miejsc kopii od razu było widać, gdzie leży kopia.

□□ ZRZUT EKRANU DO UZUPEŁNIENIA

Zrzut do uzupełnienia: ekran Settings → System → Storage → Add network storage z wypełnionym formularzem (serwer, udział, użytkownik) i wybranym użyciem Backup.

□□ **SMB** = Server Message Block: protokół udostępniania plików i drukarek w sieci Windows, na nim stoi zwykły udział sieciowy · **OS** = Operating System: system operacyjny · **NFS** = Network File

System: udostępnianie katalogów po sieci w świecie Linuksa · **CIFS** = Common Internet File System: starsza nazwa protokołu współdzielenia plików Windows, dziś częściej nazywanego SMB, pojawia się w wynikach poleceń FreeIPA dotyczących zaufania z Active Directory

KLIKAMY W INTERFEJSIE

Harmonogram kopii: Settings, System, Backups, Backup settings

```
Settings → System → Backups → Backup settings
```

Włączamy kopię automatyczną, codziennie, w czasie wybranym przez system albo własnym, i dodajemy `bkp01` jako miejsce zapisu obok miejsca lokalnego [1]. W polu liczby kopii wybieramy, ile ich zostaje, a starsze usuwa Home Assistant. Zielona zaczyna od siedmiu kopii (nasz wybór, do skorygowania po pierwszym miesiącu), a katalog z mediami i folder wspólny wyłączamy z kopii, bo dokumentacja zaleca to dla zmniejszenia rozmiaru [1].

ZMIENIA SYSTEM

Kopia kontenera: tar i scp

```
docker compose --file /docker/compose/homeassistant/compose.yaml stop
sudo tar --create --gzip --file=/tmp/homeassistant-NazwaDaty.tar.gz --
directory=/docker/apps homeassistant
docker compose --file /docker/compose/homeassistant/compose.yaml start
scp /tmp/homeassistant-NazwaDaty.tar.gz nazwa_usera@bkp01:/backup/homeassistant/
```

Wariant dla Home Assistant w kontenerze na `docker01`. Zatrzymanie kontenera na czas pakowania zapewnia spójną bazę, a przerwa trwa kilkanaście sekund. Gotowy skrypt z cykliczną wysyłką, jak dla innych aplikacji z `docker01`, opisuje strona Kopia zapasowa aplikacji dockerowych.

docker = narzędzie do kontenerów · **tar** = tape archive: archiwum plików (kiedyś na taśmę) · **scp** = secure copy: kopiowanie plików przez SSH

KLIKAMY W INTERFEJSIE

Odtworzenie kopii na nowej maszynie: ekran powitalny

```
Ekran powitalny nowej instalacji → Restore from backup
```

Kopię można wczytać już podczas pierwszego uruchomienia, a docelowy sprzęt nie musi być taki sam jak ten, z którego kopia pochodzi [1]. Potrzebujemy pliku `.tar` z `bkp01` i klucza z Vaultwardena. Test odtworzenia robimy na testowej maszynie wirtualnej co kwartał, a wynik zapisujemy w dzienniku zmian.

□□ ZMIENIA SYSTEM

Aktualizacja Home Assistant z kopią: ha core update

```
ha core update --backup
```

Polecenie robi kopię i dopiero potem aktualizuje [3]. Wersję można wskazać wprost, na przykład `ha core update --version 2026.9.4 --backup`, co przydaje się przy cofnięciu do poprzedniej wersji [3]. W interfejsie odpowiada temu Settings → System → Updates.

□□ TYLKO ODCZYT

Zmiana kanału wydań: ha supervisor options

```
ha supervisor options --channel beta
```

Kanał beta daje wersje testowe przed wydaniem. Na `ha01` zostajemy na kanale stable, a beta oglądamy najwyżej na testowej maszynie [3].

□□ TYLKO ODCZYT

Aktualizacja kontenera: zmiana tagu obrazu

```
docker compose --file /docker/compose/homeassistant/compose.yaml pull
docker compose --file /docker/compose/homeassistant/compose.yaml up --detach
```

Przed poleceniami zmieniamy numer wersji w `image:` w pliku Compose, jak opisuje strona Instalacja Home Assistant. Wcześniej robimy kopię z karty o kontenerze.

□□ **docker** = narzędzie do kontenerów

□□ PLIK KONFIGURACYJNY

Zaufanie do serwera pośredniczącego: sekcja http

```
http:
  use_x_forwarded_for: true
  trusted_proxies:
    - 10.20.10.42
  ip_ban_enabled: true
  login_attempts_threshold: 5
```

Bez tych opcji Home Assistant odrzuca żądania od serwera pośredniczącego błędem 400 [4]. Adres 10.20.10.42 to `docker01`, na którym stoi nginx. Zmiany w sekcji `http` wymagają restartu Home Assistant, a poprawność sprawdzamy poleceniem `ha core check`.

□□ ZMIENIA SYSTEM

Blok server dla Home Assistant w nginx: /docker/apps/nginx/conf.d/ha.zielona.example.conf

```
map $http_upgrade $connection_upgrade {
    default upgrade;
    ''      close;
}

server {
    listen 443 ssl;
    server_name ha.zielona.example;

    ssl_certificate     /etc/nginx/ssl/ha.zielona.example/fullchain.pem;
    ssl_certificate_key /etc/nginx/ssl/ha.zielona.example/privkey.pem;
    ssl_protocols       TLSv1.2 TLSv1.3;

    proxy_buffering off;

    location / {
        proxy_pass http://10.20.10.54;
        proxy_http_version 1.1;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
```

```
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection $connection_upgrade;
    }
}
```

Blok `map` obsługuje nagłówek `Upgrade`, bez którego nie działa WebSocket, a `proxy_buffering off` zapobiega opóźnieniom [5]. Plik przeszedł `nginx -t` w wersji 1.24. Certyfikat i klucz kładziemy tak, jak opisuje strona TLS w Nginx, a bloki `proxy_pass` jak na stronie Reverse proxy i równoważenie obciążenia. Gdy Home Assistant stoi w kontenerze na `docker01`, adres w `proxy_pass` zmieniamy na `http://10.20.10.42:8123`.

WebSocket = stałe, dwukierunkowe połączenie przeglądarki z serwerem, przez które Home Assistant na żywo odświeża panele; serwer pośredniczący musi przekazywać nagłówki Upgrade i Connection · **TLS** = Transport Layer Security: szyfrowanie połączeń, np. w HTTPS

TYLKO ODCZYT

Sprawdzenie i przeładowanie nginx: docker exec

```
docker exec nginx nginx -t
docker exec nginx nginx -s reload
```

Pierwsze polecenie sprawdza składnię, drugie przeładowuje konfigurację bez przerywania połączeń. Nazwa kontenera `nginx` pochodzi ze strony o instalacji nginx na `docker01`.

docker = narzędzie do kontenerów

KLIKAMY W INTERFEJSIE

Adresy dla aplikacji mobilnej: Settings, System, Network

```
Settings → System → Network → Home Assistant URL
```

Jako adres wewnętrzny wpisujemy `http://10.20.10.54`, a jako zewnętrzny `https://ha.zielona.example`. Aplikacja pozwala połączenia nieszyfrowane tylko w sieci domowej, więc poza biurem potrzebuje adresu HTTPS [6]. Po VPN telefon rozwiązuje `ha.zielona.example` na `docker01`, tak jak komputer w biurze.

HTTPS = HTTP zabezpieczony szyfrowaniem TLS · **VPN** = Virtual Private Network: szyfrowany tunel do sieci firmowej

KLIKAMY W INTERFEJSIE

Włączenie uwierzytelniania dwuskładnikowego: Profile, Security

```
Profil użytkownika → Security → Multi-factor Authentication Modules → Enable
```

Wybieramy moduł TOTP: aplikacja uwierzytelniająca pokazuje sześciocyfrowy kod, który zmienia się co około 30 sekund [7]. Sekret traktujemy jak hasło, a zegar `ha01` musi być dokładny, bo TOTP zależy od czasu [7].

TOTP = Time based One Time Password: jednorazowy kod zmieniający się co chwilę, generowany przez aplikację uwierzytelniającą jako jedna z metod dwuetapowego logowania

Z życia wzięte

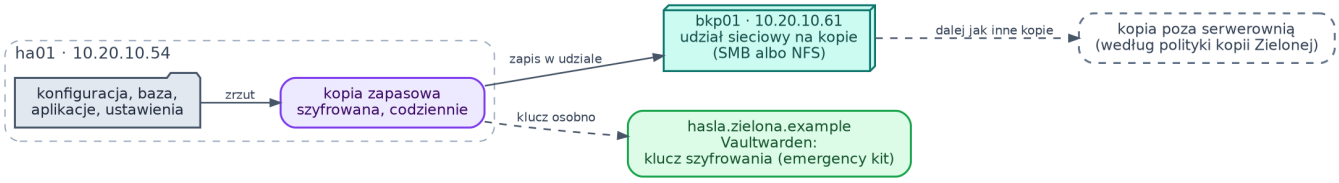
Z ŻYCIA WZIĘTE

Piątek, 15:40. Ola chce pokazać panelowi przy telefonie szefa magazynu, że działa „podgląd serwerowni”, więc prosi o otwarcie portu 8123 z internetu na `fw01`, „tylko na chwilę”. Tomek otwiera port i zapomina go zamknąć. W poniedziałek pan Wojtek, audytor, skanuje adres publiczny biura i pyta, czemu w internecie widać stronę logowania systemu do sterowania domem i serwerownią. Port znika z `fw01` w ciągu minuty, a zespół ustala, że dostęp z zewnątrz idzie tylko przez VPN.

Jak to działa

Rozwiń: jak to działa pod spodem

Co kopiujemy i gdzie to leży



Rysunek 1. Kopia zapasowa Home Assistant: zaszyfrowany zrzut konfiguracji, bazy, aplikacji i ustawień powstaje codziennie na ha01 i trafia do udziału sieciowego na bkp01, a klucz szyfrowania ląduje osobno w Vaultwardenie. Dalej kopia z bkp01 wędruje poza serwerownię jak inne kopie Zielonej. (kliknięcie otwiera pełny rozmiar)

> **Pojęcie: emergency kit.** Dokument z kluczem szyfrowania kopii Home Assistant, który pobieramy raz i trzymamy poza kopiami. Bez niego zaszyfrowanej kopii nie odtworzymy, nawet mając sam plik [1].

Zasada trzech miejsc: żywy system na `ha01`, kopia na `bkp01` i trzecia kopia poza serwerownią, a klucz w czwartym miejscu. Dokumentacja zaleca trzymać jedną kopię na innym systemie, a najlepiej też jedną poza siedzibą [1]. Kopię z `bkp01` wysyłamy dalej według polityki kopii Zielonej, tak samo jak inne.

Kopia aplikacji w kontenerze zawiera cały katalog `/docker/apps/homeassistant`, razem z bazą, plikami YAML i katalogiem `.storage`. Pliki YAML trzymamy też w repozytorium Git i nie traktujemy tej kopii jako jedynej. Baza historii bywa duża, więc przed pierwszą kopią sprawdzamy jej rozmiar i wolne miejsce w ustawieniach systemu [1].

Aktualizacje

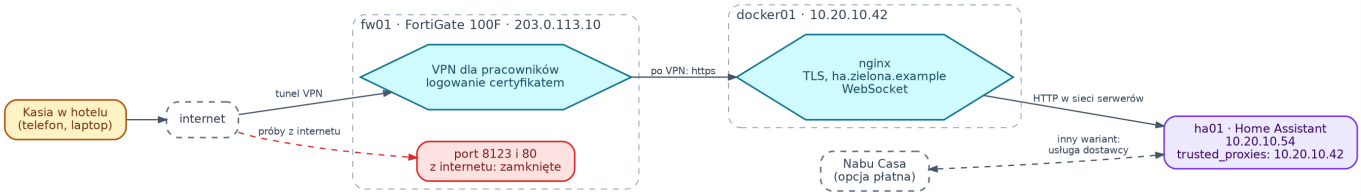
Home Assistant wydaje nową wersję co miesiąc, zwykle w pierwszą środę, a po drodze wychodzą poprawki [8]. Przed aktualizacją czytamy notatki wydania, bo każda większa wersja zawiera zmiany, które psują starsze integracje (przykład to zmiana kluczy w wyzwalaczach od 2026.7) [3][9].

Zasada	Jak robimy w Zielonej
Kanał	stable na `ha01`, beta tylko na testowej maszynie
Moment	nie w dniu wydania: czekamy na pierwszą poprawkę (nasza propozycja do potwierdzenia przez autora)
Kopia	zawsze przed aktualizacją, `ha core update --backup` [3]
Okno	poza zakazem zmian od 15 listopada do 6 stycznia

Zasada	Jak robimy w Zielonej
Cofnięcie	wersja wskazana w poleceniu `--version` albo odtworzenie kopii [3]
Płytki ESPHome	aktualizujemy osobno, po kolei, jedną płytkę na raz

Aktualizacja ESPHome nie jest aktualizacją Home Assistant. Płytki aktualizujemy po jednej, bo uszkodzona aktualizacja zdalna wymaga kabla USB i dostępu do szafy.

Dostęp zdalny



Rysunek 2. Dostęp zdalny: Kasia z hotelu łączy się tunelem VPN z fw01, a dalej po https do nginx na docker01, który przekazuje ruch HTTP do ha01. Porty 8123 i 80 z internetu zostają zamknięte (czerwona przerywana linia). Nabu Casa jest płatnym wariantem alternatywnym. (kliknięcie otwiera pełny rozmiar)

Reguła jest jedna: portu Home Assistant nie wystawiamy do internetu. Panel steruje światłami, bramami i lampą serwerowni, więc jest celem skanerów, a jedyną barierą byłoby hasło. Zostajemy przy VPN, który w Zielonej działa na `fw01` jako IPsec IKEv2 z logowaniem certyfikatem, jak opisują strony IPsec i IKEv2 po ludzku i EAP-TLS i FSSO.

Po połączeniu z VPN telefon otwiera `https://ha.zielona.example`. Nazwa wskazuje na nginx na `docker01`, który kończy TLS i przekazuje ruch HTTP do `ha01`. Home Assistant ufa tylko adresowi nginx jako serwera pośredniczącego (`trusted_proxies`) i dzięki temu widzi prawdziwe adresy klientów, a nie adres nginx [4][5].

Porównanie dróg dostępu pokazuje, czemu wybieramy VPN.

Droga	Co dostajemy	Wada
Port 8123 do internetu	prostota	cel skanerów, jedyną ochroną jest hasło, nie robimy tego
VPN na `fw01` i nginx w sieci serwerów	dostęp ze znanego mechanizmu i szyfrowanie	wymaga uruchomionego VPN w telefonie

Droga	Co dostajemy	Wada
Nabu Casa (płatna subskrypcja Home Assistant Cloud)	dostęp zdalny bez własnego VPN i nginx	opłata; ruch przez usługę dostawcy; cena do potwierdzenia na stronie dostawcy

Usługa Home Assistant Cloud to opcjonalna subskrypcja partnera projektu, firmy Nabu Casa, która dodaje bezpieczny dostęp zdalny [10]. Strona dostawcy oferuje 31 dni okresu próbnego, a aktualnej ceny nie podajemy, bo wymaga potwierdzenia u dostawcy [11]. W Zielonej to opcja zapasowa, jeżeli VPN nie wystarczy, a dla domu wygodny wariant, gdy rodzina nie ma VPN.

Konta i uwierzytelnianie

Pierwszy użytkownik utworzony na ekranie powitalnym jest właścicielem i tworzy kolejnych [12]. Zakładamy drugie konto administratora zespołu, żeby utrata jednego nie zablokowała dostępu, a zwykłe konta dla osób, które tylko patrzą na panele, jak opisuje strona Panele w Home Assistant.

W profilu każdego użytkownika włączamy TOTP. Moduł Notify wysyła kod powiadomieniem, ale nie wie, czy kod dotarł, więc przy braku powiadomienia nie możemy się zalogować [7]. Dlatego dla kont administratorów wybieramy TOTP, a Notify tylko wtedy, gdy mamy inną drogę wejścia.

Opcja `ip_ban_enabled` blokuje adres po kilku nieudanych logowaniach, a próg ustawia `login_attempts_threshold` (u nas 5) [4]. Za nginx adres w blokadzie to adres klienta dopiero po poprawnym ustawieniu `trusted_proxies`, inaczej zablokowalibyśmy adres nginx i wszystkich razem.

Blok `auth_providers`, jeśli kiedykolwiek go dodamy, musi zostać w głównym `configuration.yaml` i zawierać `type: homeassistant` jako awaryjny sposób logowania, bo bez niego grozi zablokowanie wszystkich kont [12][13]. Dostawcy `trusted_networks` nie używamy: pozwala wejść z zaufanej sieci bez hasła, a moduł dwuskładnikowy w takim logowaniu nie bierze udziału [12].

W domu

W domu zasady są te same, tylko prostsze. Kopię robimy na dysk sieciowy (NAS), a drugą kopię na zewnętrzny dysk, który nie stoi na stałe w tej samej szafie. Dostęp spoza domu to VPN na routerze albo usługa Nabu Casa, a portu 8123 nie otwieramy w routerze.

Gdy ta sama rodzina korzysta z aplikacji w telefonie, ustawiamy adres wewnętrzny na domowy, a zewnętrzny na adres HTTPS. W aplikacji wybieramy poziom, który zezwala na połączenia nieszyfrowane tylko w sieci domowej [6].

> Więcej możliwości albo gotowe rozwiązanie od ręki (out of the box): gotowy sprzęt Home Assistant z dyskiem NVMe, który zawiera system i działa od pierwszego włączenia, albo mały serwer NAS z kopią migawek i drugim zasilaniem. W dużych centrach danych stosuje się kopie według zasady 3-2-1, replikację między lokalizacjami i okresowe, zautomatyzowane testy odtworzenia, a dostęp administracyjny przez bastiony z uwierzytelnianiem wieloskładnikowym.

Częste błędy

Objaw	Przyczyna	Co robimy
Nie możemy odtworzyć kopii	brak klucza szyfrowania	klucz z emergency kit w Vaultwardenie, a po utracie tworzymy nową kopię z nowym kluczem [1]
Zapis na `bkp01` kończy się błędem uprawnień	brak etykiety SELinux `samba_share_t` albo zła własność katalogu	`semanage fcontext` i `restorecon`, a właścicielem katalogu jest `hakopie` [2]
Home Assistant nie widzi udziału `ha-kopie`	zapora na `bkp01` albo `fw01`, zła nazwa udziału	sprawdzamy `firewall-cmd --list-services`, nazwę w `smb.conf` i `testparm`
Strona przez nginx zwraca 400	brak `use_x_forwarded_for` albo zły adres w `trusted_proxies`	wpisujemy adres nginx (10.20.10.42) w sekcji `http` i restartujemy [4]
Interfejs ładuje się, ale ciągle „łączy”	brak nagłówków `Upgrade` i `Connection` w nginx	blok `map` i dwa nagłówki z fiszki [5]
Po aktualizacji integracja przestała działać	zmiana niezgodna wstecz	cofamy wersją z `--version` albo odtwarzamy kopię, a w notatkach wydania szukamy zmiany [3]
Zablokowano nasz adres po kilku próbach logowania	`ip_ban_enabled` i zła hasła	czekamy na zdjęcie blokady albo usuwamy wpis z pliku `ip_bans.yaml` w katalogu konfiguracji i restartujemy
Aplikacja mobilna nie łączy się spoza biura	brak VPN w telefonie albo adres zewnętrzny bez HTTPS	włączamy VPN i sprawdzamy adres zewnętrzny [6]
Po dodaniu `auth_providers` nikt nie może się zalogować	brak `type: homeassistant`	przywracamy plik konfiguracji z kopii i dopisujemy dostawcę [13]

Ćwiczenia

1. Dlaczego klucz szyfrowania kopii trzymamy w Vaultwardenie, a nie obok kopii na `bkp01`?

Odpowiedź

Kopia i klucz w tym samym miejscu znikają razem, a kto zdobyłby to miejsce, miałby i kopię, i klucz. Osobny klucz chroni przed jednym i drugim [1].

1. Ola pyta, czy w nginx wystarczy `proxy_pass`. Co jeszcze jest potrzebne, żeby Home Assistant działał przez serwer pośredniczący?

Odpowiedź

W nginx blok `map` z nagłówkami `Upgrade` i `Connection` dla WebSocket, a w Home Assistant sekcja `http` z `use_x_forwarded_for` i `trusted_proxies` z adresem nginx. Bez drugiej części Home Assistant odrzuca żądania błędem 400 [4][5].

1. Aktualizacja do nowej wersji zepsuła integrację, na której zależy panel serwerowni. Co robimy po kolei?

Odpowiedź

Cofamy wersję poleceniem `ha core update --version NumerWersji --backup` albo odtwarzamy kopię sprzed aktualizacji, a potem czytamy notatki wydania i czekamy na poprawkę [3]. W Zielonej nie aktualizujemy w okresie zakazu zmian od 15 listopada do 6 stycznia.

1. Prezes chce otwierać panel z hotelu. Którą drogę wybieramy i czego nie robimy?

Odpowiedź

Wybieramy VPN na `fw01` i adres `https://ha.zielona.example`. Nie otwieramy portu 8123 ani 80 na `fw01`, nawet tymczasowo, bo skanery znajdują taki port szybko.

Powiązane: Instalacja Home Assistant: system na maszynie wirtualnej, mini PC albo w kontenerze · Automatyzacje i powiadomienia: alarm temperatury w serwerowni bez dublowania Zabbixa · Panele w Home Assistant: karty, panel serwerowni, panel domowy i uprawnienia · Reverse proxy i równoważenie obciążenia · Kopia zapasowa aplikacji dockerowych · Pierwsza konfiguracja Vaultwardena

Źródła:

1. Home Assistant, kopie zapasowe (szyfrowanie, emergency kit, lokalizacje, harmonogram, odtwarzanie przy pierwszym uruchomieniu): home-assistant.io/common-tasks/general
2. samba_selinux(8), etykieta `samba_share_t`: mankier.com/8/samba_selinux
3. Home Assistant, zadania w systemie (magazyn sieciowy NFS i CIFS, `ha core update --backup`, `--version`, kanały): home-assistant.io/common-tasks/os
4. Home Assistant, integracja HTTP (`trusted_proxies`, `use_x_forwarded_for`, `ip_ban_enabled`, błąd 400): home-assistant.io/integrations/http
5. Społeczność Home Assistant, konfiguracja nginx (`map`, `Upgrade`, `proxy_buffering off`): community.home-assistant.io/t/reverse-proxy-using-nginx/196954
6. Home Assistant Companion, uruchomienie aplikacji (adres wewnętrzny i zewnętrzny, połączenia nieszyfrowane tylko w domu): companion.home-assistant.io/docs/getting_started
7. Home Assistant, uwierzytelnianie wieloskładnikowe (TOTP, moduł Notify): home-assistant.io/docs/authentication/multi-factor-auth
8. Home Assistant 2026.9, notatki wydania (miesięczny rytm wydań, wydanie z 2 września 2026): home-assistant.io/blog/2026/09/02/release-20269
9. Home Assistant 2026.7, notatki wydania (zmiany nazw kluczy): home-assistant.io/blog/2026/07/01/release-20267
10. Home Assistant, Home Assistant Cloud: home-assistant.io/cloud
11. Nabu Casa, strona usługi (31 dni okresu próbnego): nabucasa.com
12. Home Assistant, dostawcy uwierzytelniania (właściciel, `trusted_networks`, `auth_providers`): home-assistant.io/docs/authentication/providers
13. Home Assistant, pakiety (`auth_providers` w głównym pliku): home-assistant.io/docs/configuration/packages

Home Assistant: częste problemy z urządzeniami, bazą i aktualizacjami

Poziom: operacyjny

Dotyczy: Home Assistant 2026.9, ESPHome 2026.9.1, Avahi, FortiGate fw01

Czas: ~45 min

Zestawienie objawów, z którymi zespół IT trafia do Home Assistant najczęściej: urządzenie `unavailable`, płytka niewidoczna w innej podsieci, rosnąca baza, aktualizacja, po której integracja przestaje działać, i ESPHome, które się nie kompiluje. Tabela w części „Częste błędy” daje szybką odpowiedź, a rozwinięcia wyżej tłumaczą przyczyny do końca. Na początku są komendy, od których zaczynamy każdą diagnozę.

Skróty użyte na tej stronie (22)

Skrót	Co znaczy
API	Application Programming Interface: sposób, w jaki programy rozmawiają ze sobą
Avahi	program dla Linuksa obsługujący mDNS; z włączonym reflektorem powtarza ogłoszenia mDNS między dwiema sieciami
CLI	Command Line Interface: praca przez wpisywanie komend w terminalu
DHCP	Dynamic Host Configuration Protocol: usługa, która sama rozdaje komputerom adresy IP
DNS	Domain Name System: książka telefoniczna internetu, zamienia nazwy na adresy IP
ESP32	Tani mikrokontroler z Wi-Fi i Bluetooth firmy Espressif: malutki komputer, z którego budujemy własne czujniki sieciowe
ESPHome	Oprogramowanie, które zamienia ESP32 w gotowy czujnik sieciowy bez pisania kodu od zera, a odczyty udostępnia przez sieć
GB	Gigabyte: miliard bajtów, jednostka pojemności pamięci i plików, tu najczęściej rozmiar modelu albo pamięci karty graficznej
IP	Internet Protocol: adres komputera w sieci; w elektryce: stopień ochrony obudowy (Ingress Protection)

Skrót	Co znaczy
IPAM	IP Address Management: dokumentacja tego, który adres IP jest zajęty przez co, żeby dwa urządzenia nie dostały tego samego adresu
IT	Information Technology: dział albo praca związana z komputerami i infrastrukturą firmy
mDNS	Multicast DNS: sposób, w którym urządzenia w jednej sieci lokalnej ogłaszają swoją nazwę (np. esp01.local) zapytaniami na adres multicast, bez serwera DNS; nie przechodzi między podsieciami
OS	Operating System: system operacyjny
PlatformIO	narzędzie do budowania programów dla mikrokontrolerów, z którego ESPHome korzysta przy kompilacji; jego pliki czyści esphome clean-all
PoE	Power over Ethernet: zasilanie urządzenia tym samym kablem sieciowym, którym płyną dane
SQLite	mała baza danych zapisana w jednym pliku; Home Assistant trzyma w niej historię w pliku home-assistant_v2.db
TCP	Transmission Control Protocol: protokół sieciowy z potwierdzaniem dostarczenia
VLAN	Virtual LAN: wydzielona logicznie sieć na tym samym sprzęcie
Wi-Fi	Sieć bezprzewodowa: wygodna w biurze, ale w serwerowni dokłada kolejną możliwą przyczynę awarii
YAML	YAML Ain't Markup Language: format plików konfiguracyjnych oparty na wcięciach
Zeroconf	inna nazwa mechanizmu samoczynnego znajdowania urządzeń w sieci lokalnej (mDNS i pokrewne); Home Assistant używa go do odkrywania urządzeń
Zigbee	Radiowy protokół sieci domowych o małym poborze prądu, używany w czujkach, czujnikach i przełącznikach na baterie

Fiszka

□□ TYLKO ODCZYT

Dziennik Home Assistant: ha core logs

```
ha core logs
```

W Home Assistant OS polecenie wykonujemy w konsoli systemu albo w aplikacji terminala. W kontenerze na `docker01` używamy `docker logs --tail 200 homeassistant`. W interfejsie ten sam dziennik jest w Settings → System → Logs, a pierwszy komunikat o błędzie integracji zwykle wyjaśnia resztę.

□□ **OS** = Operating System: system operacyjny

TYLKO ODCZYT

Sprawdzenie portu API płytki: nc

```
nc -zv NazwaPlytki 6053
```

Polecenie wykonujemy z `docker01` albo ze stacji administratora w tej samej sieci co `ha01`. Komunikat o udanym połączeniu oznacza, że płytka nasłuchuje na porcie API 6053 i nic w drodze go nie blokuje. Brak połączenia przy działającym `ping` wskazuje zaporę albo płytkę, na której działa tylko `web_server`.

API = Application Programming Interface: sposób, w jaki programy rozmawiają ze sobą

TYLKO ODCZYT

Dziennik płytki przez sieć: esphome logs

```
esphome logs esp01.yaml
```

Dziennik pokazuje ponowne łączenia z siecią, restarty i klientów API, więc od razu widać, czy płytka restartuje się co 15 minut.

API = Application Programming Interface: sposób, w jaki programy rozmawiają ze sobą · **esphome** = ESPHome: buduje i wgrywa oprogramowanie na ESP32 z pliku YAML, pokazuje logi urządzenia

TYLKO ODCZYT

Rozmiar bazy Home Assistant: ls

```
ls -lh /config/home-assistant_v2.db
```

Domyślna baza SQLite to plik `home-assistant_v2.db` w katalogu konfiguracji [1]. W kontenerze na `docker01` ten plik leży w `/docker/apps/homeassistant/`. Rozmiar bazy i zajęte miejsce widzimy też w Settings → System → Storage.

SQLite = mała baza danych zapisana w jednym pliku; Home Assistant trzyma w niej historię w pliku home-assistant_v2.db · **ls** = list: wypisz zawartość katalogu

PLIK KONFIGURACYJNY

Ustawienia recorder w configuration.yaml: recorder

```
recorder:
  purge_keep_days: 14
  commit_interval: 10
  exclude:
    domains:
      - update
    entity_globs:
      - sensor.*_sila_sygnalu*
      - sensor.*_czas_pracy*
    event_types:
      - call_service
```

Domyślnie baza trzyma 10 dni historii, czyszczenie startuje co noc o 4:12, a pełne przepakowanie robi się w drugą niedzielę miesiąca po czyszczeniu. Domyślny `commit_interval` to 5 sekund [1]. Wykluczamy to, czego nie oglądamy na wykresach: moc sygnału Wi-Fi, czas pracy płytek i zdarzenia wywołań akcji. Po zmianie sekcji `recorder` restartujemy Home Assistant, a poprawność sprawdzamy poleceniem `ha core check`.

□□ **Wi-Fi** = Sieć bezprzewodowa: wygodna w biurze, ale w serwerowni dokłada kolejną możliwą przyczynę awarii

□□ PLIK KONFIGURACYJNY

Jednorazowe czyszczenie bazy: recorder.purge

```
action: recorder.purge
data:
  keep_days: 7
  repack: true
```

Akcję uruchamiamy w Settings → Tools → Actions, wybierając YAML i wklejając blok. Pole `keep_days` mówi, ile dni zostawiamy, a `repack` zmniejsza plik po usunięciu danych [1]. Przepakowanie obciąża dysk, więc robimy je poza godzinami, w których `ha01` jest oglądany.

□□ **YAML** = YAML Ain't Markup Language: format plików konfiguracyjnych oparty na wcięciach

□□ ZMIENIA SYSTEM

Cofnięcie do poprzedniej wersji: ha core update

```
ha core update --version 2026.9.3 --backup
```

Numer wersji wpisujemy ten, który działał, na przykład 2026.9.3, a opcja `--backup` robi kopię przed zmianą [2]. W kontenerze zmieniamy tag w pliku Compose i robimy `pull` oraz `up`, jak na stronie Instalacja Home Assistant.

TYLKO ODCZYT

Wyczyszczenie plików kompilacji ESPHome: esphome clean

```
esphome clean esp01.yaml
```

Polecenie czyści pliki kompilacji jednej płytki i pomaga przy części błędów budowania [3]. Gdy to nie pomaga, `esphome clean-all` czyści pliki wszystkich płytek, pakiety i rdzeń PlatformIO [3]. Po `clean-all` pierwsza kompilacja trwa dłużej, bo biblioteki pobierają się od nowa.

ESPHome = Oprogramowanie, które zamienia ESP32 w gotowy czujnik sieciowy bez pisania kodu od zera, a odczyty udostępnia przez sieć · **PlatformIO** = narzędzie do budowania programów dla mikrokontrolerów, z którego ESPHome korzysta przy kompilacji; jego pliki czyści esphome clean-all · **esphome** = ESPHome: buduje i wgrywa oprogramowanie na ESP32 z pliku YAML, pokazuje logi urządzenia

TYLKO ODCZYT

Wolne miejsce na dysku kompilacji: df

```
df -h /
```

Kompilacja ESPHome potrzebuje miejsca na pliki tymczasowe i biblioteki. Na stacji administratora sprawdzamy dysk domowy użytkownika, a w kontenerze ESPHome wolumen `/docker`.

ESPHome = Oprogramowanie, które zamienia ESP32 w gotowy czujnik sieciowy bez pisania kodu od zera, a odczyty udostępnia przez sieć · **df** = disk free: ile miejsca wolnego na dyskach

PLIK KONFIGURACYJNY

Odbicie mDNS między sieciami: avahi-daemon.conf

```
[server]
allow-interfaces=eth0,eth1

[reflector]
enable-reflector=yes
```

Avahi z włączonym reflektorem powtarza zapytania mDNS na wszystkie dozwolone interfejsy [4]. Nazwy `eth0` i `eth1` to przykład dwóch interfejsów w dwóch sieciach i podstawiamy własne. Po zmianie restartujemy usługę poleceniem `sudo systemctl restart avahi-daemon`.

mDNS = Multicast DNS: sposób, w którym urządzenia w jednej sieci lokalnej ogłaszają swoją nazwę (np. esp01.local) zapytaniami na adres multicast, bez serwera DNS; nie przechodzi między podsieciami · **Avahi** = program dla Linuksa obsługujący mDNS; z włączonym reflektorem powtarza ogłoszenia mDNS między dwiema sieciami

Z życia wzięte

Z ŻYCIA WZIĘTE

Wtorek, 7:10. Marek montuje w chłodni w Swarzędzu płytkę ESP32 z czujnikiem temperatury, ta sama jak w serwerowni, tylko na Wi-Fi w VLAN 30. W Home Assistant na `ha01` płytka się nie pojawia, choć świeci i ma adres. Tomek szuka godzinę, aż Ola pyta, czy `ha01` i płytka są w tej samej sieci: nie są, a mDNS nie przechodzi przez `fw01`, więc dodają płytkę ręcznie po adresie IP i otwierają jeden port.

Jak to działa

Rozwiń: jak to działa pod spodem

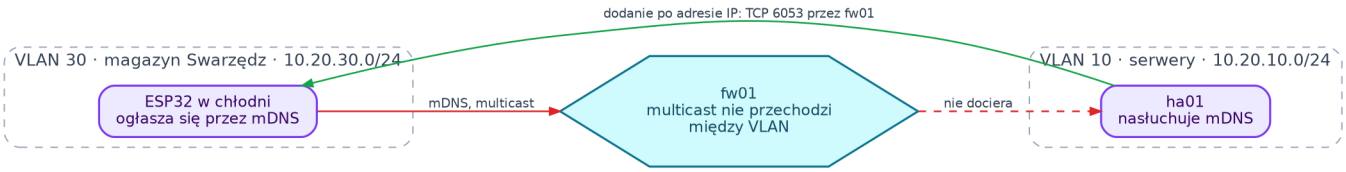
Dlaczego urządzenie jest unavailable

Pojęcie: stan unavailable. Stan encji, którego Home Assistant nie może odczytać, bo urządzenie lub integracja nie odpowiada. Różni się od `unknown`, który znaczy, że encja istnieje, ale jeszcze nie ma wartości.

Płytka z natywnym API ma stałe połączenie, więc `unavailable` pojawia się zaraz po jego zerwaniu. Przyczyna to zasilanie, sieć, zmiana klucza szyfrowania, restart płytki albo zmiana adresu IP. Przy dodaniu przez nazwę `esp01.local` płytka zależy od mDNS, a przy dodaniu przez adres od niezmienności adresu, więc dla płytek ustawiamy statyczny adres, jak na stronie o ESPHome.

mDNS i podsieci

> **Pojęcie: mDNS.** Mechanizm, w którym urządzenia ogłaszają się nazwą w sieci lokalnej zapytaniami wysyłanymi na adres multicast, bez serwera DNS. Zapytania mDNS zostają w jednej sieci, bo routery nie przekazują ich między podsieciami.



Rysunek 1. Płytka ESP32 w chłodni w VLAN 30 ogłasza się przez mDNS, ale zapytania multicast nie przechodzą przez fw01 do ha01 w VLAN 10 (czerwona linia). Dodanie po adresie IP otwiera połączenie TCP 6053 z ha01 do płytki przez fw01 (zielona linia). (kliknięcie otwiera pełny rozmiar)

Mamy dwie drogi. Pierwsza to dodanie płytki ręcznie po adresie, który przy dodaniu ESPHome przyjmuje zamiast nazwy `.local` [5]. Na `fw01` otwieramy wtedy tylko połączenie TCP z `ha01` na port 6053 płytki, a płytce nadajemy stały adres z rezerwacji DHCP.

Druga droga to reflektor mDNS, na przykład Avahi na hoście, który ma interfejsy w obu sieciach. Reflektor rozszerza ogłoszenia na dodatkowe sieci, ale wymaga dodatkowego hosta i jego utrzymania [4]. FortiGate może mieć własną funkcję przekazywania mDNS, ale jej dostępność w naszej wersji FortiOS i ustawienia sprawdzamy w dokumentacji producenta, do potwierdzenia. Dokumentacja Home Assistant nie opisuje mDNS między podsieciami, więc w razie wątpliwości wybieramy dodanie po adresie [6].

Dla płytek, które mają być w innym VLAN-ie, wybieramy dodanie po adresie. Odkrywanie jest wygodne, ale nie daje się łatwo dokumentować i przestaje działać po zmianie sieci.

Baza rośnie

Home Assistant zapisuje każdą zmianę stanu każdej encji. Czujnik, który co 30 sekund podaje nową wartość, daje do 2880 zapisów dziennie, a sto takich czujników ponad 288 tysięcy. Plik bazy rośnie wtedy w gigabajty, a kopia zapasowa i start systemu zwalniają.

Mamy trzy narzędzia. Skracamy `purge_keep_days`, wykluczamy encje, których historii nie oglądamy (`exclude`), i w razie potrzeby uruchamiamy jednorazowe `recorder.purge`. Dane do wykresów długoterminowych zostają w statystykach długoterminowych, które recorder trzyma osobno, więc skrócenie historii nie zabiera wykresów z kilku miesięcy.

Dokumentacja zaznacza, że SQLite jest najlepiej przetestowaną bazą, a wybór innej wymaga umiejętności administrowania bazą i pilnowania miejsca na dysku [1]. W Zielonej zostajemy przy SQLite. Gdy baza i tak rośnie, `pg01` z PostgreSQL jest opcją: wystarczy `db_url` w sekcji `recorder` w postaci `postgresql://user:password@SERVER_IP/DB_NAME` [1]. Przy zmianie bazy historia ze starej nie przenosi się sama, więc przed przełączeniem robimy kopię.

Aktualizacja zepsuła integrację

Każda większa wersja Home Assistant może zmienić integrację niezgodnie ze starą konfiguracją, na przykład 2026.7 zmieniło nazwy części kluczy wyzwalaczy [7]. Dlatego przed aktualizacją robimy kopię, czytamy notatki wydania i czekamy na pierwszą poprawkę, jak na stronie o kopii i aktualizacji. Gdy po aktualizacji coś nie działa, najpierw dziennik, potem notatki wydania, a dopiero potem cofnięcie wersji [2].

ESPHome się nie kompiluje

Najczęstsze przyczyny to błąd składni w pliku, brak miejsca na dysku, zbyt stary Python i uszkodzone pliki kompilacji. Składnię sprawdza `esphome config`, wersję Pythona wymaga pakiet (3.12 do 3.14 dla wersji 2026.9.1), a pliki kompilacji czyści `esphome clean` [3]. Gdy kompilacja zatrzymuje się na pobieraniu bibliotek, sprawdzamy dostęp stacji do internetu przez serwer pośredniczący.

W domu

W domu najczęstszy problem to router, który nie przepuszcza mDNS między Wi-Fi a siecią kablową albo ma włączoną izolację klientów Wi-Fi. Płytka wtedy świeci, ma adres, a Home Assistant jej nie widzi. Pierwszy krok jest ten sam: dodajemy płytkę ręcznie po adresie, a w routerze rezerwujemy dla niej stały adres.

> Więcej możliwości albo gotowe rozwiązanie od ręki (out of the box): sieciowy czujnik z PoE w jednym VLAN-ie z serwerem, który nie wymaga odkrywania, albo gotowy hub Zigbee z własną siecią radiową, który nie zależy od mDNS. W dużych centrach danych stosuje się osobne sieci zarządzania urządzeń z jawnie opisanymi regułami zapory i inwentarzem adresów w systemie typu IPAM, więc nikt nie polega na rozgłaszaniu.

Częste błędy

Objaw	Przyczyna	Co robimy
Encja płytki ma stan `unavailable`	płytka bez zasilania, zerwane połączenie z siecią, zmieniony adres albo klucz API	`ping` i `nc -zv NazwaPlytki 6053`, potem `esphome logs`, a przy zmianie klucza wpisujemy nowy w integracji
Płytka restartuje się co 15 minut, a po restarcie wraca	`api` bez klienta i domyślny `reboot_timeout`	`reboot_timeout: 0s` w sekcji `api`, jeśli płytkę czyta też Zabbix [8]
Płytka nie pojawia się w Discovered	mDNS nie przechodzi między podsieciami albo jest wyłączony	dodajemy ręcznie po adresie IP, port 6053, a na `fw01` otwieramy TCP z `ha01` do płytki [5]
Po dodaniu po adresie płytka znów jest `unavailable`	adres się zmienił, bo był z DHCP bez rezerwacji	rezerwacja DHCP albo statyczny adres w pliku płytki
Baza `home-assistant_v2.db` ma kilka gigabajtów	długa historia i dużo encji, które zmieniają się często	`purge_keep_days`, `exclude`, jednorazowo `recorder.purge` z `repack: true` [1]
Po `recorder.purge` plik się nie zmniejszył	usunięte dane nie zmniejszają pliku bez przepakowania	akcja z `repack: true` albo czekamy na przepakowanie w drugą niedzielę miesiąca [1]
Po zmianie bazy na PostgreSQL nie ma starej historii	historia nie przenosi się sama	przed zmianą robimy kopię i zakładamy, że historia zaczyna się od zera [1]
Po aktualizacji integracja przestała działać	zmiana niezgodna wstecz w nowej wersji	dziennik, notatki wydania, a potem `ha core update --version ... --backup` albo odtworzenie kopii [2][7]
`esphome` zgłasza błąd przy kompilacji	składnia pliku, brak miejsca, niewłaściwy Python	`esphome config`, `df -h`, `python3 --version`, potem `esphome clean` albo `esphome clean-all` [3]
`homeassistant.local` się nie otwiera	mDNS nie działa albo nowa instalacja słucha na porcie 80, nie 8123	wchodzimy po adresie IP i sprawdzamy port w ustawieniach sieci

Objaw	Przyczyna	Co robimy
Strona za nginx zwraca 400	brak `trusted_proxies`	patrz strona Kopia, aktualizacja i dostęp zdalny
Nasz adres zablokowany po próbach logowania	`ip_ban_enabled` po zbyt wielu błędach	czekamy albo usuwamy wpis z `ip_bans.yaml` i restartujemy

□□ Ćwiczenia

1. Płytka w VLAN 30 świeci i ma adres, a w Home Assistant jej nie ma w Discovered. Co robimy najpierw?

Odpowiedź

Sprawdzamy `ping` i `nc -zv NazwaPlytki 6053` z sieci `ha01`, a gdy port odpowiada, dodajemy płytkę ręcznie po adresie IP. Brak odkrywania wynika z tego, że mDNS nie przechodzi przez `fw01` [5].

1. Plik `home-assistant_v2.db` ma 6 GB. Co ustawiamy, żeby baza przestała rosnąć tak szybko?

Odpowiedź

Skracamy `purge_keep_days`, wykluczamy encje i zdarzenia, których historii nie oglądamy, a na koniec uruchamiamy `recorder.purge` z `repack: true`, żeby zmniejszyć plik [1].

1. Czy po zmianie bazy na PostgreSQL na `pg01` zostają nam wykresy z ostatniego roku?

Odpowiedź

Nie z samej zmiany: historia ze starej bazy nie przenosi się sama, więc przed przełączeniem robimy kopię i liczymy się z początkiem historii od zera [1].

1. Aktualizacja do nowej wersji zepsuła integrację w piątek o 15:50. Czy cofamy wersję od razu?

Odpowiedź

Najpierw patrzymy w dziennik i notatki wydania, bo często wystarczy poprawić jedno ustawienie. Gdy panel serwerowni ma działać, cofamy wersję poleceniem z `--version` i `--backup` albo odtwarzamy kopię sprzed aktualizacji [2]. Poprawkę robimy w poniedziałek, a w okresie zakazu zmian (od 15 listopada do 6 stycznia) w ogóle nie aktualizujemy.

Powiązane: Instalacja Home Assistant: system na maszynie wirtualnej, mini PC albo w kontenerze · ESPHome i urządzenia: adopcja esp01 i esp02, API albo MQTT, nazwy encji · Kopia, aktualizacja i dostęp zdalny: Home Assistant bez otwartego portu 8123 · Serwerownia Zielonej: gdzie stoją czujniki i dokąd płyną alarmy

Źródła:

1. Home Assistant, integracja Recorder (`purge_keep_days`, `commit_interval`, `exclude`, `recorder.purge`, `db_url`): home-assistant.io/integrations/recorder
2. Home Assistant, zadania w systemie (`ha core update --backup`, `--version`): home-assistant.io/common-tasks/os
3. ESPHome, polecenia CLI (`config`, `compile`, `run`, `logs`, `clean`, `clean-all`): esphome.io/guides/cli
4. Avahi, avahi-daemon.conf(5), sekcja `[reflector]`: manpages.debian.org/bookworm/avahi-daemon/avahi-daemon.conf.5.en.html
5. ESPHome, Native API Component (dodanie przez nazwę `.local` albo adres IP): esphome.io/components/api
6. Home Assistant, integracja Zeroconf: home-assistant.io/integrations/zeroconf
7. Home Assistant 2026.7, notatki wydania (zmiany nazw kluczy): home-assistant.io/blog/2026/07/01/release-20267
8. ESPHome, Native API Component (`reboot_timeout`): esphome.io/components/api