



BridgeLink WebAdmin Setup Guide

Installation, Deployment, and First Login

CONFIDENTIAL: For authorized BridgeLink WebAdmin use only.
© 2026 Innovar Healthcare. All rights reserved.

Table of Contents

About This Guide3

System Requirements4

Deployment Models7

Installing BridgeLink WebAdmin9

Configuration Reference18

Running BridgeLink WebAdmin as a Background Service23

Logging In26

Upgrading27

Troubleshooting28

Support30

About This Guide

BridgeLink WebAdmin is the browser-based administration interface for BridgeLink. It runs as a small server process alongside your BridgeLink server and lets any administrator manage channels, messages, users, and settings from a modern browser, with no per-workstation client install. This guide is for the person installing and deploying WebAdmin (an IT administrator or BridgeLink administrator), and it covers system requirements, installing WebAdmin on Linux, macOS, Windows, and Docker, choosing a deployment model, configuring TLS and access control, running WebAdmin as a background service, and logging in for the first time.

System Requirements

WebAdmin has two components with separate resource profiles: the **application server** (a small Node.js process that serves the UI and proxies API calls to your BridgeLink server) and each user's **browser** (where the UI actually renders).

BridgeLink server compatibility: WebAdmin requires BridgeLink server version 26.3.0 or later. WebAdmin checks the server version at login and blocks connection to a server older than the minimum; a server newer than this build of WebAdmin shows a dismissible notice instead of blocking.

Server Requirements

The application server does not cache channel data between sessions, so memory stays nearly flat as users connect.

Resource	Idle	Per active user	Recommended (production)
RAM	~65 MB	+5 to 10 MB	2 GB (supports ~10 concurrent users)
CPU	Less than 1%	Mostly proxy overhead	2 vCPU
Disk	500 MB	None	1 GB (room for logs and the application)
Open file descriptors	Low	~5 per active session	Default system limits are sufficient

For larger deployments, scale linearly. At approximately 50 concurrent users, target 4 vCPU and 4 GB RAM. The bottleneck is network egress to browsers, not CPU or RAM.

Supported server operating systems:

- **Linux:** Node.js 22.x LTS; RHEL 8+, Ubuntu 20.04+, Debian 11+, Amazon Linux 2/2023, SLES 15+.
- **macOS:** Node.js 22.x LTS. Supported for local testing or single-user use; not recommended as a production server install.
- **Windows:** Windows 10 or later (64-bit), including Windows Server editions built on the same platform version. This floor comes from the Windows installer itself, which requires 64-bit Windows 10 or later and refuses to run on anything older.
- **Docker:** any Linux host with Docker 20.10+ (RHEL 8+, Ubuntu 20.04+, Debian 11+, Amazon Linux 2/2023, SLES 15+), image platforms [linux/amd64](#) and [linux/arm64](#).

Network traffic from the application server to each browser, with gzip compression enabled by default:

Traffic	Direction	Volume
Initial channel load (one time per browser session)	UI to browser	~1.5 MB
Dashboard status poll (every 20 seconds by default, while the Dashboard tab is open)	UI to browser	~35 KB
API calls to BridgeLink	UI to BridgeLink	Varies, passthrough

A user with the Dashboard tab open for an 8 hour workday consumes approximately 50 MB of bandwidth from the application server at the default poll interval. The interval is configurable under **Settings**, on the **Administrator** tab, and bandwidth scales inversely with it.

Browser Requirements

Each user runs the UI in their own browser. The application server does not share state between users.

Resource	Minimum	Recommended
RAM available to the browser tab	500 MB	1 GB
Browser	Chrome 120, Edge 120, Firefox 120, Safari 17	Latest stable release
Network to the application server	5 Mbps	Broadband
Screen resolution	1280 x 800	1920 x 1080

A Channels tab loaded against a 200 to 300 channel deployment uses approximately 250 to 400 MB of browser tab memory. The channel editor loads on first use and adds roughly 50 MB.

Licensing

BridgeLink WebAdmin is licensed under the Business Source License 1.1 (BSL 1.1) with the BridgeLink WebAdmin Supplemental Terms.

Read both documents in full before deploying WebAdmin in production. They are published at the BridgeLink Web UI GitHub repository, <https://github.com/Innovar-Healthcare/BridgeLink-Web-UI>:

- **LICENSE**, the Business Source License 1.1
- **SUPPLEMENTAL-TERMS.md**, the BridgeLink WebAdmin Supplemental Terms

Contact Innovar Healthcare for commercial licensing questions.

License acceptance during install or startup is a behavior of the installers and the Docker image, not a summary of what the license permits:

- The **Windows and Linux/macOS installers** each present the Business Source License 1.1 followed by the BridgeLink WebAdmin Supplemental Terms and require you to accept both before installation proceeds (see Installing BridgeLink WebAdmin below). A silent installer run (`/SILENT` or `/VERYSILENT` on Windows, `BL_ACCEPT_LICENSE=1` on Linux/macOS) accepts both documents implicitly, the same posture most silent installers take.
- The **Docker image** enforces license acceptance at container startup (see Installing BridgeLink WebAdmin, Docker, below).
- The **manual tarball and zip artifacts** (described under the advanced installation paths below) do not gate startup on a license-acceptance step.

License acceptance is an install-time or startup gate only; it is never written to `webadmin.conf`, which holds runtime configuration exclusively.

Deployment Models

Before installing, decide how WebAdmin will reach your BridgeLink server or servers. This choice determines which environment variable (or `webadmin.conf` entry) you set, and it changes what the login screen looks like.

Model	How it is configured	Login experience
Same host (default)	Leave both <code>BRIDGELINK_SERVER_URL</code> and <code>BRIDGELINK_ALLOWED_SERVERS</code> unset. Set <code>BRIDGELINK_PUBLIC_HOST</code> to WebAdmin's public hostname (or <code>host:port</code>).	The Server field is pre-filled with the same-host BridgeLink server and is read-only.
Single server	<code>BRIDGELINK_SERVER_URL=https://your-bridgelink-server:8443</code>	Login locks to that one server; the Server field shows it pre-filled and read-only.
Multi-server console	<code>BRIDGELINK_ALLOWED_SERVERS=https://bl-a:8443,https://bl-b:8443</code> (comma-separated, two or more)	The Server field becomes a dropdown so the user chooses which BridgeLink server to manage.

The Windows and Linux/macOS installers ask for this choice directly during setup and write the matching configuration for you, under these on-screen names: **Same Host (Recommended)** for Same host, **Dedicated Host – Single Server** for Single server, and **Dedicated Host – Multi-Server** for Multi-server console. See Installing BridgeLink WebAdmin below.

A few points apply across all three models:

- WebAdmin's proxy relays requests only to servers it is explicitly configured to allow (an SSRF guard). Matching is exact scheme, host, and port, and only `https://` targets are accepted.
- In **Same host** mode, the allowed BridgeLink server is derived from the browser's **Host** header unless `BRIDGELINK_PUBLIC_HOST` is set. Setting `BRIDGELINK_PUBLIC_HOST` is recommended whenever you are not using an explicit allowlist, so this derivation is based on a trusted value rather than a header a client could spoof.
- **Same host** mode always targets the BridgeLink server at port **8443** on that host (BridgeLink's default HTTPS port); this port is fixed and does not follow whatever port WebAdmin itself listens on, and any port included in `BRIDGELINK_PUBLIC_HOST` is ignored for this purpose. If your BridgeLink server listens on a different port, use the **Single server** model with an explicit `BRIDGELINK_SERVER_URL` instead.
- Set `BRIDGELINK_SERVER_URL` or `BRIDGELINK_ALLOWED_SERVERS`, not both; if both happen to be set, `BRIDGELINK_ALLOWED_SERVERS` (multi-server console) takes precedence.

A development-only escape hatch exists and must never be used in production.

BL_ALLOW_ANY_SERVER=1 disables the server allowlist entirely and restores a free-text **Server** field for any host. It is intended for local development only (it is also enabled automatically whenever NODE_ENV=development) and must never be set on a shared or internet-facing deployment, because it removes the SSRF protection described above.

Installing BridgeLink WebAdmin

WebAdmin is delivered in four installable forms, built from the same codebase. For a new Windows or Linux/macOS deployment, use the installer: a single downloadable package that installs the application, generates a TLS certificate, writes `webadmin.conf` from the answers you give it, and registers a background service, in one guided pass. Docker and the manual archives remain available as alternative paths.

Format	Best for
Windows installer (.exe)	Windows servers. The recommended Windows path. Bundles the Node.js runtime, so no internet connection is required during setup.
Linux/macOS installer (<code>install.sh</code> , shipped inside a <code>.tar.gz</code>)	Linux or macOS servers. The recommended Linux/macOS path. Automates Node.js installation, TLS certificate generation, and service registration.
Docker image	Linux servers already running Docker, and multi-user deployments.
Manual archives (raw tarball or zip)	Restrictive network policies, hosts where you do not want an installer-managed service, or environments already standardized on your own service-management tooling.

All WebAdmin packages are available from bridgelink.net/download. WebAdmin is offered both bundled with the BridgeLink server download and as a separate WebAdmin-only download; the packages are the same either way, so choose whichever fits how you are installing BridgeLink.

macOS runs the same installer and the same manual tarball as Linux (see below); both are supported only for local testing or single-user use, not as a production server.

Docker

The Docker path is recommended for Linux servers that already run Docker and for multi-user deployments where you want the container runtime to manage the service lifecycle.

Two image variants are published to Docker Hub:

Image	Contents
<code>innovarhealthcare/bridgelink-webadmin</code>	Base build, open-source plugins only
<code>innovarhc/bridgelink-web-ui-advanced</code>	Base plus commercial plugins (SSL Settings, OIDC, Access Control, and so on)

Only the base image has been renamed to the `innovarhealthcare/` namespace; the commercial (advanced) image still publishes under its former `innovarhc/` name, pending a separate rename decision.

Both share the same tag scheme: `:latest`, a pinned version such as `:26.6.0` (recommended for production), a dated build such as `:20260720`, or an exact commit such as `:sha-b6a7884`. Images are multi-arch (`linux/amd64` and `linux/arm64`, native builds).

1. Create a `docker-compose.yml` file:

`services:`

```
bridgelinek-webui:
  image: innovarhealthcare/bridgelinek-webadmin:latest
  container_name: bridgelinek-webui
  ports:
    - "8444:8444"
  environment:
    PORT: "8444"
    NODE_ENV: production
    BRIDGELINK_SERVER_URL: https://your-bridgelinek-server:8443
    # Required: accepts the BSL 1.1 license and the BridgeLink WebAdmin
    # Supplemental Terms. See Licensing above.
    BL_ACCEPT_LICENSE: "1"
  volumes:
    - bridgelinek-certs:/app/certs
  restart: unless-stopped
```

`volumes:`

```
bridgelinek-certs:
```

2. Start the container:

```
docker compose up -d
```

3. Watch the logs to confirm a clean start:

```
docker compose logs -f
```

4. Open `https://localhost:8444` in a browser (or whatever host port you mapped).

BL_ACCEPT_LICENSE=1 is required. Omitting this variable, or setting it to any value other than 1, prints the license terms to the container log and exits immediately. Existing deployments upgrading from an image that predates this requirement must add the variable before pulling a newer image, or the container will restart-loop.

TLS. The container serves HTTPS by default. On first start, the entrypoint checks for a certificate and key at `/app/certs/server.crt` and `/app/certs/server.key`. If both are present, they are used as is. If either is missing, the entrypoint generates a fresh self-signed certificate with a unique private key (825-day validity); each container instance gets its own key. Mount a Docker named volume at `/app/certs` (as shown above) so an auto-generated certificate survives container recreation, or bind-mount your own certificate:

```
docker run -d \  
  -p 8444:8444 \  
  -e PORT=8444 \  
  -v /etc/ssl/bridgelink:/app/certs:ro \  
  -e BRIDGELINK_SERVER_URL=https://your-server:8443 \  
  -e BL_ACCEPT_LICENSE=1 \  
  innovarhealthcare/bridgelink-webadmin:latest
```

The container runs as the non-root **node** user (UID 1000, GID 1000). A bind-mounted host directory must be readable by UID 1000, and writable too if you want an auto-generated certificate to persist back to the host:

```
sudo chown -R 1000:1000 /etc/ssl/bridgelink
```

Docker named volumes inherit correct ownership automatically and need no **chown**. The container also cannot bind to host ports below 1024 from inside itself; map the host port instead, for example **-p 443:8444**.

Updating:

```
docker compose pull && docker compose up -d
```

To roll back, pin an earlier tag (for example **innovarhealthcare/bridgelink-webadmin:26.6.0**) in **docker-compose.yml** and run **docker compose up -d** again.

Linux and macOS (installer, recommended)

Download **BridgeLink-WebAdmin-<version>-install.tar.gz** from bridgelink.net/download. Extract it and run its **install.sh**:

```
tar -xzf BridgeLink-WebAdmin-<version>-install.tar.gz  
cd bridgelink-webadmin  
sudo bash install.sh
```

If you run it as a non-root user, **install.sh** re-executes itself with **sudo** automatically; you do not need to add **sudo** yourself.

1. **License Agreement.** The script points to the Business Source License 1.1 (**LICENSE**) and the BridgeLink WebAdmin Supplemental Terms (**SUPPLEMENTAL-TERMS.txt**), both shipped next to **install.sh**, then prompts **Do you accept the license terms? [y/N]:**. Any answer other than **y** cancels the installation.
2. **Node.js.** If Node.js is not already installed, you can auto-install it (via **apt-get** on Ubuntu/Debian, **dnf** on Amazon Linux 2023, Fedora, or RHEL 8+, **yum** on Amazon Linux 2 or CentOS 7, or Homebrew on macOS) or point the installer at an existing installation.
3. **File copy.** The script copies itself to **/opt/bridgelink-webadmin/** and installs the application's Node.js dependencies.
4. **Port.** Enter the HTTPS port WebAdmin will listen on, or press Enter to accept the default:

Enter the port for BridgeLink WebAdmin [default: 8444]:

5. **Deployment model.** Choose which deployment model this WebAdmin instance uses:

Prompt option	Deployment model
1) Same Host (Recommended)	Same host
2) Dedicated Host - Single Server	Single server (prompts for one server URL next)
3) Dedicated Host - Multi-Server	Multi-server console (prompts for two or more server URLs next; minimum two)

See Deployment Models above for what each model writes to `webadmin.conf`.

After the prompts, the script:

- Writes `/opt/bridgelinek-webadmin/webadmin.conf` with `PORT=` and, unless you chose Same Host, either `BRIDGELINK_SERVER_URL=` or `BRIDGELINK_ALLOWED_SERVERS=`.
- Generates a self-signed TLS certificate valid for 10 years at `/opt/bridgelinek-webadmin/cert.pem` and `key.pem`, and appends their paths to `webadmin.conf` as `SSL_CERT_FILE=` and `SSL_KEY_FILE=`.
- On Linux, writes a systemd unit to `/etc/systemd/system/bridgelinek-webadmin.service` and starts it immediately. On macOS, writes a LaunchDaemon plist to `/Library/LaunchDaemons/com.innovar.bridgelinek-webadmin.plist` and loads it.
- Installs a small management wrapper at `/usr/local/bin/blwebadmin`:

```
blwebadmin start | stop | restart | status | logs
```

When installation finishes, open `https://localhost:<port>` (or `https://<server-address>:<port>` from another machine).

Uninstalling. Run `sudo bash install.sh --uninstall` from the extracted package. This stops and removes the service, deletes `/opt/bridgelinek-webadmin/`, and removes the `blwebadmin` CLI.

Upgrading. Re-running `install.sh` does not currently seed its prompts from your existing `webadmin.conf`. It asks for the port and deployment model again and generates a fresh TLS certificate, so re-run it with the same answers you used originally (or use the unattended variables below to make the re-run reproducible), and expect the certificate to be replaced.

Unattended install. Supply answers ahead of time instead of answering the prompts interactively, using any combination of:

| Variable | Default | Description |

| --- | --- | --- |

| BL_ACCEPT_LICENSE | None | Set to 1 to accept the license terms. Required for --non-interactive; interactive installs prompt instead. |

| BL_PORT | 8444 | HTTPS port WebAdmin listens on. |

| BL_SERVER_URL | None | Single BridgeLink server URL (<https://...>). Takes precedence over BL_SERVER_URLS if both are set. |

| BL_SERVER_URLS | None | Comma-separated list of two or more server URLs (<https://...>). |

| BL_SAME_HOST | None | Set to 1 for Same Host mode; no server URL is written. Cannot be combined with BL_SERVER_URL or BL_SERVER_URLS; the installer exits with an error if you set both. |

| NODEJS_HOME | None | Directory containing bin/node, if Node.js is not on PATH. |

Set these as environment variables (run `sudo -E bash install.sh` rather than plain `sudo` so they survive the switch to root), or copy `install-answers.conf.template` (shipped next to `install.sh`) to `install-answers.conf` and uncomment the values you need; the config file does not require `-E`. Add `--non-interactive` for CI or pipeline use: it exits with an error instead of prompting for anything still missing, and requires `BL_ACCEPT_LICENSE=1` plus one of the deployment-model variables.

Linux and macOS (manual tarball, advanced path)

Use the manual tarball instead of the installer if you want to run WebAdmin without an installer-managed systemd unit or LaunchDaemon, for example to wire it into your own service-management tooling or a container-orchestration layer, or in an environment where you would rather not auto-install Node.js system-wide. This path does not gate on a license-acceptance step and does not register a service for you (see Running BridgeLink WebAdmin as a Background Service below to do that yourself).

Download the tarball from bridgelink.net/download. The Linux and macOS artifact is the same `.tar.gz` tarball, launched with the same `start-https.sh` script. Node.js does not need to be installed first: the launcher downloads Node.js 22 automatically on first run if it is not already present (an internet connection is required only for that first run).

1. Extract the tarball to any directory on the machine:

```
tar -xzf bridgelink-webadmin.tar.gz -C /opt/
cd /opt/bridgelink-webadmin
```

2. If you want to set the BridgeLink server URL before the first launch, edit `webadmin.conf` in this folder, or export it as an environment variable:

```
export BRIDGELINK_SERVER_URL=https://your-bridgelink-server:8443
```

3. Start the server:

`./start-https.sh`

- The script downloads Node.js 22 if needed, generates a self-signed TLS certificate in `certs/` if one is not already present, and starts the HTTPS server, printing **Ready on https://localhost:8444** (or your configured port) when it is up.
- Certificate trust (optional).** The launcher also offers to add the self-signed certificate to your system trust store so your browser does not show a security warning:

Platform	Mechanism
macOS	Login keychain (<code>security add-trusted-cert</code>); no sudo needed
Linux	RHEL-family (<code>update-ca-trust</code>) or Debian-family (<code>update-ca-certificates</code>), depending on your distribution; asks for <code>sudo</code>

Answering **N** at the prompt is fine; your browser will simply ask you to confirm the certificate once.

- Open **https://localhost:8444** (or your configured port) in a browser.

To use your own certificate instead of the auto-generated one, place `server.crt` and `server.key` in the `certs/` folder before running `start-https.sh` for the first time; the script detects existing files and skips generation.

To use a custom port:

`PORT=8443 ./start-https.sh`

macOS-specific note. If OpenSSL is not already installed, `start-https.sh` attempts to install it automatically: via Homebrew if Homebrew is present, or by prompting you to install the Xcode Command Line Tools if it is not. Both require confirming an on-screen prompt.

Windows (installer, recommended)

Download `BridgeLink-WebAdmin-<version>-Setup.exe` from bridgelink.net/download and run it. It requires 64-bit Windows 10 or later and around 120 MB of free disk space, and requests administrator elevation automatically.

- License Agreement.** Review and accept the Business Source License 1.1.
- Supplemental Terms.** Review and accept the BridgeLink WebAdmin Supplemental Terms on a second page. Both documents require acceptance to continue; declining either cancels the installation.
- Select Destination Folder.** Defaults to `C:\Program Files\BridgeLink WebAdmin\`.
- Select Additional Tasks.** Two optional checkboxes:
 - Create a desktop shortcut** (unchecked by default)
 - Install as Windows Service (auto-start at boot)** (checked by default)

5. **Port Configuration.** Enter the HTTPS port WebAdmin will listen on. The field is pre-filled with **8444**. Valid range is 1 to 65535.
6. **BridgeLink Server.** Choose a deployment model:

Option	Deployment model
Same Host (Recommended)	Same host
Dedicated Host – Single Server	Single server (the next page prompts for that server's URL)
Dedicated Host – Multi-Server	Multi-server console (the next page prompts for two or more server URLs)

See Deployment Models above for what each model configures.

7. If you chose **Dedicated Host – Single Server** in the previous step, a page prompts for that server's URL. It must start with **https://** and include a hostname.
8. If you chose **Dedicated Host – Multi-Server**, a page lets you add server URLs:
 - Type a URL and click **Add**, repeating until every server is listed
 - Click **Remove Selected** to delete an entry

At least two URLs are required to continue.

9. **Ready to Install.** Review your choices, then click **Install**.

After the files are copied, the installer writes `{app}\webadmin.conf` with `PORT=` and, depending on the model chosen in step 6, either `BRIDGELINK_SERVER_URL=` or `BRIDGELINK_ALLOWED_SERVERS=`. Choosing **Same Host** writes no server-authorization line at all: WebAdmin auto-authorizes the BridgeLink installation on the same host without one. If **Install as Windows Service** was checked, the installer also:

- Installs and starts a Windows Service named **BridgeLinkWebAdmin**, set to auto-start at boot.
- Generates a self-signed TLS certificate (or reuses one already present) at `C:\ProgramData\BridgeLink\certs\server.crt` and `server.key`, valid for `localhost` and the machine's computer name, and adds it to the Local Machine Root certificate store so browsers on that machine do not show a warning.
- Adds a Windows Firewall inbound rule named **BridgeLink WebAdmin HTTPS** for the port you chose.

When installation finishes, open WebAdmin from the **BridgeLink WebAdmin** shortcut in the Start Menu (or the optional desktop shortcut). With the service installed, this shortcut opens `https://<computer-name>:<port>` directly; without it, the shortcut runs `launch-webadmin.bat`, which starts the application and then opens your browser.

Silent installs. Running the installer with `/SILENT` or `/VERYSILENT` skips the wizard pages and implicitly accepts both license documents, the standard convention for silent Windows installers.

Upgrading. Running the installer again over an existing installation carries your previous port, deployment model, and server URLs forward automatically; you are not asked to re-enter them unless you want to change something. If a Windows Service from a previous install is still registered and you leave **Install as Windows Service** unchecked this time, the installer asks whether to remove it.

Windows (manual zip, advanced path)

Use the manual zip instead of the installer if you need to run WebAdmin without an installer-managed Windows Service, for example to wire it into your own service-management tooling, or to test it without an administrator installing anything system-wide. This path does not gate on a license-acceptance step and does not register a service for you (see Running BridgeLink WebAdmin as a Background Service below to do that yourself).

Download the zip from bridgelink.net/download.

1. Extract the zip file to any directory on the machine, for example with File Explorer's **Extract All**, or:

```
Expand-Archive bridgelink-webadmin.zip -DestinationPath C:\bridgelink-webadmin
```

2. If you want to set the BridgeLink server URL before the first launch, edit `webadmin.conf` in the extracted folder, or set the environment variable first:

```
$env:BRIDGELINK_SERVER_URL = "https://your-bridgelink-server:8443"
```

3. Double-click `start-https.bat` in Explorer, or run it from Command Prompt or PowerShell:

```
.\start-https.bat
```

4. The script downloads Node.js 22 if needed (stored under `%LOCALAPPDATA%\BridgeLink\.node`, so no administrator rights are required), generates a self-signed TLS certificate if needed, and starts the HTTPS server.
5. The script also offers to add the self-signed certificate to your Windows certificate store (**Cert:\CurrentUser\Root**) so your browser does not show a security warning. Answering **N** is fine; your browser will simply ask you to confirm the certificate once.
6. Open `https://localhost:8444` (or your configured port) in a browser.

Certificate location is different on Windows. `start-https.bat` stores and looks for its TLS certificate under `%LOCALAPPDATA%\BridgeLink\certs\`, not in the `certs\` folder inside the extracted zip. To use your own certificate on Windows, create that folder (if it does not already exist) and place `server.crt` and `server.key` there before running `start-https.bat`.

To use a custom port:

```
set PORT=8443
.\start-https.bat
```

Important: always run `start-https.bat`, not `start-https.ps1` directly. Running the `.ps1` file directly can fail with an "Unauthorized Access" error unless your system's script execution policy allows unsigned scripts; the `.bat` file bypasses that automatically.

Configuration Reference

webadmin.conf: the configuration file

WebAdmin reads its configuration from a single file, `webadmin.conf`, a plain `KEY=VALUE` file that ships next to `server.js` in every deployed artifact (the Docker image, the Linux/macOS tarball, and the Windows zip all include one; the Windows and Linux/macOS installers write it for you). The application whitelists exactly four keys from this file: `PORT`, `BRIDGELINK_SERVER_URL`, `BRIDGELINK_ALLOWED_SERVERS`, and `BRIDGELINK_PUBLIC_HOST`. Any other line in `webadmin.conf` is ignored by the application itself (see Environment variables below for how the remaining settings reach the process).

How `webadmin.conf` gets populated depends on how you deployed WebAdmin:

Deployment	How <code>webadmin.conf</code> is populated
Docker	Overwritten at container start by the entrypoint script, from the <code>environment:</code> values in your <code>docker-compose.yml</code> (or <code>-e</code> flags).
Linux/macOS installer	Written by <code>install.sh</code> from your answers to the port and deployment-model prompts.
Windows installer	Written by the setup wizard from your answers to the port and deployment-model pages.
Linux/macOS manual tarball	Edit the file directly after extracting the tarball, or export the equivalent environment variables before running <code>start-https.sh</code> .
Windows manual zip	Edit the file directly after extracting the zip, or set the equivalent environment variables before running <code>start-https.bat</code> .

Editing `webadmin.conf`, or any of the environment variables below, requires a restart of the server process (or the service, if you registered one; see Running BridgeLink WebAdmin as a Background Service below) to take effect.

Environment variables

Every setting beyond the four keys above, `SSL_CERT_FILE`, `SSL_KEY_FILE`, `BRIDGELINK_CA_CERT`, `COOKIE_SECURE`, `TRUST_PROXY`, and `BETA_MODE`, must be a real operating-system environment variable. The application never reads these from `webadmin.conf`, even if a line for one is present there.

How each deployment type supplies these environment variables:

Deployment	How environment variables reach the process
Docker	The <code>environment:</code> block in <code>docker-compose.yml</code> (or <code>-e</code> flags on <code>docker run</code>).
Linux (installer, systemd)	systemd's <code>EnvironmentFile=</code> directive loads every line in <code>webadmin.conf</code> as a real environment variable for the process, not just the four keys <code>server.js</code> parses on its own. This is why the Linux installer's own <code>webadmin.conf</code> also carries <code>SSL_CERT_FILE=</code> and <code>SSL_KEY_FILE=</code> lines even though the application does not read those two keys from the file directly: systemd is what actually turns those lines into environment variables.
macOS (installer, LaunchDaemon)	The installer's service wrapper script reads <code>webadmin.conf</code> and exports each line as an environment variable before starting the process, the same effect as systemd's <code>EnvironmentFile=</code> on Linux.
Windows (installer, NSSM service)	The installer does not route extra keys through <code>webadmin.conf</code> at all: it points the Windows Service directly at a fixed certificate path (<code>C:\ProgramData\BridgeLink\certs\server.crt</code> and <code>server.key</code>) and never writes <code>SSL_CERT_FILE/SSL_KEY_FILE</code> into <code>webadmin.conf</code> .
Manual tarball or zip (no service registered)	Export the variables in your shell (or your own service manager's environment configuration) before starting <code>start-https.sh</code> / <code>start-https.bat</code> , or before running <code>node server.js</code> directly.

Core variables

Variable	Where it is set	Default	Description
PORT	<code>webadmin.conf</code> or environment	8444	TCP port the application listens on. Both installers and the shipped <code>webadmin.conf</code> set 8444.
BRIDGELINK_SERVER_URL	<code>webadmin.conf</code> or environment	None	Single-server deployment model. See Deployment Models above.
BRIDGELINK_ALLOWED_SERVERS	<code>webadmin.conf</code> or environment	None	Multi-server console deployment model (comma-separated). See Deployment Models above.

Variable	Where it is set	Default	Description
BRIDGELINK_PUBLIC_HOST	webadmin.conf or environment	None	Trusted public hostname (or host:port) of this WebAdmin instance, used in Same host mode. See Deployment Models above.
SSL_CERT_FILE	Environment only	None (Docker: /app/certs/server.crt)	Path to the TLS certificate PEM file for WebAdmin.
SSL_KEY_FILE	Environment only	None (Docker: /app/certs/server.key)	Path to the TLS private key PEM file for WebAdmin.
BRIDGELINK_CA_CERT	Environment only	None	Absolute path to a PEM file (CA or self-signed certificate) for the Web Admin-to-BridgeLink hop. See TLS below.
COOKIE_SECURE	Environment only	Auto-detected	Force the session cookie's Secure flag and HSTS on (true) or off (false). Leave unset to auto-detect based on whether the request arrived over a secure context.
TRUST_PROXY	Environment only	true	Set to false to stop trusting the X-Forwarded-Proto header from a reverse proxy, which disables HTTPS detection through that proxy.
BETA_MODE	Environment only	Unset	When set to true, shows a pre-release beta disclaimer at login that users must accept. Leave unset for a general-availability deployment.

TLS

WebAdmin serves HTTPS whenever both `SSL_CERT_FILE` and `SSL_KEY_FILE` are set (the launcher scripts and the Docker entrypoint set these automatically). If either is missing, the

server falls back to plain HTTP with a startup warning, which is the expected configuration only when a reverse proxy in front of WebAdmin terminates TLS instead.

Every installation path generates a self-signed certificate automatically the first time no certificate is found, so HTTPS works out of the box; the file names and location depend on which path you used:

Path	Certificate location
Windows installer	C:\ProgramData\BridgeLink\certs\server.crt and server.key
Windows manual zip	%LOCALAPPDATA%\BridgeLink\certs\server.crt and server.key
Linux/macOS installer	/opt/bridgelink-webadmin/cert.pem and key.pem
Linux/macOS manual tarball	certs/server.crt and certs/server.key, inside the extracted tarball's own folder
Docker	/app/certs/server.crt and server.key (mount a volume here to persist it, see Docker above)

Because it is self-signed, browsers show a one-time warning; click **Advanced**, then **Proceed** (Chrome or Edge), or **Accept the Risk and Continue** (Firefox). For a multi-user production deployment, replace it with a certificate from a trusted CA at the applicable location above, or place WebAdmin behind a reverse proxy that terminates TLS with a CA-issued certificate instead.

The WebAdmin-to-BridgeLink hop is a separate TLS relationship from the browser-to-WebAdmin one above. By default, a self-signed certificate on your BridgeLink server is accepted automatically, so a typical single-server or lab deployment needs no extra configuration. If you want full certificate verification on this hop instead, set `BRIDGELINK_CA_CERT` to the absolute path of the CA (or self-signed) certificate that signed your BridgeLink server's certificate.

Do not set `NODE_TLS_REJECT_UNAUTHORIZED`. It is no longer used by WebAdmin and disables TLS verification for the entire Node.js process, not just the connection to your BridgeLink server. If you are seeing a network error at login and suspect a certificate problem, use `BRIDGELINK_CA_CERT` instead (self-signed BridgeLink certificates are already accepted automatically without any configuration).

Behind a reverse proxy or load balancer

If you terminate TLS at an external reverse proxy (nginx, an AWS Application Load Balancer, and so on) instead of using WebAdmin's built-in TLS, run the server without `SSL_CERT_FILE` or `SSL_KEY_FILE` set and let the proxy forward plain HTTP to WebAdmin's port. Confirm the proxy sends **X-Forwarded-Proto: https** so WebAdmin's cookie and HSTS logic detects the secure context correctly (this is the default expectation; set `TRUST_PROXY=false` only if you

intentionally want to ignore that header). Example nginx configuration:

```
server {
    listen 443 ssl;
    server_name bridgelink.example.com;

    ssl_certificate      /etc/ssl/certs/bridgelink.crt;
    ssl_certificate_key  /etc/ssl/private/bridgelink.key;

    location / {
        proxy_pass http://localhost:8444;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        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;
    }
}
```

Running BridgeLink WebAdmin as a Background Service

Running `start-https.sh` or `start-https.bat` directly in a terminal is convenient for testing, but the process stops when the terminal closes. For a persistent server deployment, register WebAdmin with your platform's service manager.

If you used an installer

Both installers register a service for you; there is nothing further to set up.

- **Windows.** If you checked **Install as Windows Service** during setup, WebAdmin already runs as the Windows Service **BridgeLinkWebAdmin**, installed and managed with NSSM behind the scenes. Manage it from **Services** (`services.msc`), or from an elevated prompt:

```
sc query BridgeLinkWebAdmin
sc stop BridgeLinkWebAdmin
sc start BridgeLinkWebAdmin
```

Service logs are written to `logs\service.log` and `logs\service-error.log` inside the install directory. To add or remove the service after the fact, run the installer again and change the **Install as Windows Service** checkbox.

- **Linux.** The installer registers and starts a systemd unit named **bridgelink-webadmin**. Manage it with the `blwebadmin` CLI it also installs, or with `systemctl` directly:

```
blwebadmin status
blwebadmin restart
journalctl -u bridgelink-webadmin -f
```

- **macOS.** The installer registers a LaunchDaemon named **com.innovar.bridgelink-webadmin**, loaded at boot and restarted automatically if it exits. Manage it with the same `blwebadmin` CLI:

```
blwebadmin status
blwebadmin restart
```

Logs are written to `/var/log/bridgelink-webadmin.log` and `.err`.

Manual path (raw tarball or zip): register a service yourself

If you installed from the manual tarball or zip instead, WebAdmin does not register a service on its own. The options below are the same regardless of how you installed; they apply whenever you are running the manual archives and want WebAdmin to survive a reboot or an accidental terminal close.

Linux (systemd, recommended)

```
# /etc/systemd/system/bridgelink-webui.service
[Unit]
Description=BridgeLink WebAdmin
```

```
After=network.target
```

```
[Service]
Type=simple
WorkingDirectory=/opt/bridgelink-webadmin
ExecStart=/opt/bridgelink-webadmin/start-https.sh
Environment=NODE_ENV=production
Restart=on-failure
User=bridgelink
```

```
[Install]
WantedBy=multi-user.target
```

```
sudo systemctl enable bridgelink-webui
sudo systemctl start bridgelink-webui
```

Check status and logs with `systemctl status bridgelink-webui` and `journalctl -u bridgelink-webui -f`.

Linux and macOS (pm2, cross-platform alternative)

```
npm install -g pm2
```

```
cd bridgelink-webui
```

```
SSL_CERT_FILE=./certs/server.crt \
SSL_KEY_FILE=./certs/server.key \
BRIDGELINK_SERVER_URL=https://your-bridgelink-server:8443 \
pm2 start server.js --name bridgelink-webui
```

```
pm2 save
pm2 startup
```

Check status with `pm2 status`, tail logs with `pm2 logs bridgelink-webui`, and stop with `pm2 stop bridgelink-webui`.

macOS is not recommended for a production server deployment. Source documentation for WebAdmin covers pm2 as above for keeping the process alive on macOS, but there is no built-in launchd or LaunchDaemon integration. For a production macOS-hosted deployment, use Docker or move the workload to Linux instead.

Windows (NSSM or node-windows)

The manual zip does not register a Windows Service on its own (only the installer does that, see above). Use a service wrapper such as [NSSM](#) or [node-windows](#) to run it as a Windows service yourself. Because these wrappers call `node.exe` directly (bypassing `start-https.bat`), install Node.js 22 yourself first, and generate or place a certificate at a known path before configuring the service, then point the wrapper at that path.

Install NSSM, then register the service:

```
nssm install BridgeLinkWebUI "C:\Program Files\nodejs\node.exe" "C:\BridgeLink\webadmin\server.js"
nssm set BridgeLinkWebUI AppDirectory "C:\BridgeLink\webadmin"
nssm set BridgeLinkWebUI AppEnvironmentExtra ^
    "NODE_ENV=production" ^
    "PORT=8444" ^
    "BRIDGELINK_SERVER_URL=https://your-bridgelink-server:8443" ^
    "SSL_CERT_FILE=C:\BridgeLink\webadmin\certs\server.crt" ^
    "SSL_KEY_FILE=C:\BridgeLink\webadmin\certs\server.key"

nssm start BridgeLinkWebUI
```

Docker

restart: `unless-stopped` in the `docker-compose.yml` example under Installing BridgeLink WebAdmin, Docker, above already keeps the container running across host reboots and container crashes; no separate service registration is needed.

Logging In

1. Open a browser and navigate to **https://<server-address>:<port>**, using the hostname or IP address of the machine running WebAdmin and the port you configured. Use **https://localhost:<port>** if you are on the same machine. WebAdmin listens on **8444** by default; both installers prompt for the port during setup and pre-fill **8444**.
2. If your browser shows a certificate warning, this is expected with the auto-generated self-signed certificate. Click **Advanced**, then **Proceed** (Chrome or Edge), or **Accept the Risk and Continue** (Firefox).
3. Review the **Server** field. In same host and single server deployments it is pre-filled with the configured BridgeLink server and is read-only. In a multi-server console it is a dropdown; select the BridgeLink server you want to manage.
4. Enter your BridgeLink username and password.
5. Click **Login**.

WebAdmin checks the BridgeLink server's version as part of login; a server older than 26.3.0 is rejected, and a server newer than this build of WebAdmin shows a dismissible notice.

Upgrading

Docker:

```
docker compose pull && docker compose up -d
```

Windows installer: run the new `BridgeLink-WebAdmin-<version>-Setup.exe` over the existing installation. It carries your previous port, deployment model, and server URLs forward automatically (see Installing BridgeLink WebAdmin, Windows, above), stops the running service before replacing files, and restarts it afterward.

Linux/macOS installer: re-running `install.sh` does not read your existing `webadmin.conf`; it asks for the port and deployment model again and generates a new TLS certificate. Have your previous answers on hand, or use the unattended-install variables (see Installing BridgeLink WebAdmin, Linux and macOS, above) to reproduce them exactly.

Manual tarball or zip (Linux, macOS, Windows):

1. Stop the running WebAdmin service or process.
2. Back up the current installation directory.
3. Extract the new build artifact over the existing directory.
4. Carry over your `webadmin.conf` and `certs/` folder from the backup.
5. Start the WebAdmin service or process again.

```
sudo systemctl stop bridgelink-webui
mv /opt/bridgelink-webadmin /opt/bridgelink-webadmin.bak
tar -xzf bridgelink-webadmin.tar.gz -C /opt/
# Carry over webadmin.conf and certs/ from the backup, then:
sudo systemctl start bridgelink-webui
```

To roll back, reverse the same steps using your backed-up directory.

Troubleshooting

Symptom	Likely cause	What to do
node: command not found, or the Node.js download fails on first launch	No internet access, or a firewall or proxy is blocking <code>nodejs.org</code>	Download the Node.js 22 LTS installer manually on a machine with internet access, install it on the target machine, then re-run the launcher script. It detects the installed version and skips the download.
EADDRINUSE: address already in use on the configured port	Another process is already listening on that port	Stop the conflicting process, or set <code>PORT</code> to a different value and restart.
Browser shows "Your connection is not private" or a certificate warning	Expected with the auto-generated self-signed certificate	Click Advanced , then Proceed (Chrome or Edge), or Accept the Risk and Continue (Firefox). For production, replace it with a CA-issued certificate or terminate TLS at a reverse proxy.
Login fails immediately with a network error	WebAdmin cannot reach the BridgeLink server	Confirm <code>BRIDGELINK_SERVER_URL</code> (or the URL entered at login) points to the correct host and port. Confirm the server is reachable, for example <code>curl -k https://10.0.0.5:8443/api/server/version</code> . Do not set <code>NODE_TLS_REJECT_UNAUTHORIZED</code> ; self-signed BridgeLink certificates are accepted automatically.
Docker container exits immediately or restart-loops	<code>BL_ACCEPT_LICENSE</code> is unset or not exactly 1	Add <code>BL_ACCEPT_LICENSE: "1"</code> to the container's environment and recreate it.
Session expires unexpectedly	Normal inactivity timeout	Log in again to resume.
Page does not load at the expected URL	The service or process is not running, or it started on a different port	Check the service status (<code>systemctl status bridgelink-webui</code> , <code>pm2 status</code> , <code>blwebadmin status</code> , or <code>sc query BridgeLinkWebAdmin</code> for an installer-registered Windows Service) and confirm the port matches the URL you are using.
Windows installer's service does not start, or the Start Menu shortcut opens an unreachable page	The Windows Service failed to start, commonly a port conflict or a Node.js runtime issue	Check <code>logs\service.log</code> and <code>logs\service-error.log</code> in the install directory. Confirm the port is not already in use, and confirm the Windows Firewall rule BridgeLink WebAdmin HTTPS exists for that port.

Symptom	Likely cause	What to do
<code>install.sh</code> exits with "License terms not accepted" or <code>--non-interactive</code> errors about <code>BL_ACCEPT_LICENSE</code>	The license prompt was answered with anything other than <code>y</code> , or <code>--non-interactive</code> was used without <code>BL_ACCEPT_LICENSE=1</code>	Re-run <code>install.sh</code> and answer <code>y</code> , or set <code>BL_ACCEPT_LICENSE=1</code> (environment variable or <code>install-answers.conf</code>) before an unattended run.
<code>blwebadmin</code> : command not found	<code>/usr/local/bin</code> is not on <code>PATH</code> , or the installer was interrupted before it finished	Confirm <code>/usr/local/bin/blwebadmin</code> exists; if not, re-run <code>install.sh</code> (safe to repeat). If it exists, add <code>/usr/local/bin</code> to your <code>PATH</code> .

Support

Contact Innovar Healthcare support at support@innovarhealthcare.com if you cannot resolve an issue with this guide. Include the WebAdmin build identifier (the image tag or the artifact filename you installed), your browser and OS, what you were trying to do, what happened versus what you expected, and any browser console errors (open DevTools, then **Console**). Innovar Healthcare's professional services team can also help plan a multi-server or high-availability WebAdmin deployment.