

HOLLYHOCK

Server Installation & Field Operations Manual

Technician Edition • Manual Installation Guide • Developer Pack • Version 1.0

From a fresh Ubuntu server to a verified server, Android field test, office review, and tested backup.

Prepared for technicians who want a complete, self-contained install path: server, database, web application, public site, Android installation, first visit, testing, backup, recovery, and troubleshooting.

Coal Road Productions

Reference platform: Ubuntu Server 24.04 LTS • Node.js 22 • MySQL 8 • Nginx • systemd

Release validation notice: release-specific commands, filenames, service units, environment keys, migration scripts, and package hashes must be checked against the verified Developer Pack release before shipment.

October 2026

Welcome to the Wonderful World of Hollyhock Server Administration

Hollyhock is a care-operations platform built to keep the daily work of a home-care organization in one place: leads, clients, offices, caregivers, schedules, family access, messages, visit evidence, CARE-GPS, hours, review, and the web tools that connect them.

This manual is written for the technician who has to make the system work. You do not need a special installation assistant. You need a fresh Ubuntu server, root or sudo access, the Hollyhock Developer Pack, DNS access for the customer domain, and enough patience to read the result of each command before moving to the next one.

The goal is simple: start with a blank Ubuntu Server 24.04 LTS machine and finish with a working Hollyhock installation where an administrator can sign in, create an office and client, assign a caregiver, install the Android app, run a real test visit, clock out, and review the visit from the office side.

What this manual does not include. Permanent customer passwords are not printed in this book or stored on the install media. A facilitator or authorized administrator supplies temporary login information. The user creates a new password during first login.

This is the technician edition. It explains what software Hollyhock uses, why it is there, what a good result looks like, and where to go when a step fails. The commands are meant to be copied exactly unless a placeholder is shown in angle brackets, such as <SERVER_IP>. Replace placeholders with the real value before you run the command.

Why this package exists

Hollyhock was built from a large amount of real development, field testing, failure analysis, and revision. The Developer Pack is meant to make that work portable. A customer can run a branded copy on infrastructure they control, with a technician following a documented path instead of rebuilding the system from memory.

For the money, the goal is straightforward: make Hollyhock one of the strongest care-operations software values available. The software should be useful on day one, understandable to the people who operate it, and repairable by a competent technician with this manual and the files in the pack.

The reference build used by this manual

Component	Reference choice	Why Hollyhock uses it
Operating system	Ubuntu Server 24.04 LTS (Noble)	Long-term support, predictable package management, strong cloud/VPS support.
Application runtime	Node.js 22.x; npm 10.x	The current Hollyhock application is built and tested on the Node 22 line.

Component	Reference choice	Why Hollyhock uses it
Core database	MySQL 8.0	Handles concurrent office, client, schedule, authorization, audit, and visit data better than a single-file database.
Public-site security state	SQLite where packaged/required	A small local database is useful for narrow replay/abuse state without making the public site own the core care database.
Web front end / reverse proxy	Nginx	Terminates HTTPS, routes public traffic to the correct local service, and keeps Node ports off the public internet.
Service manager	systemd	Starts Hollyhock at boot, restarts failed services, runs timers, and gives technicians one place to inspect status/logs.
TLS certificates	Let's Encrypt + Certbot	Provides normal browser-trusted HTTPS and automated certificate renewal.
File transfer	SFTP/SCP over SSH	Encrypted transfer using the same secure administrative channel. Plain FTP is not recommended.

Current database note. The accepted Hollyhock server baseline uses MySQL 8. Do not replace MySQL with MariaDB just because the commands look similar. Use the database engine named by the release you are installing.

Release and Manual Compatibility Record

Record the exact software and mobile versions this copy of the manual was validated against. Do not hand a technician a manual that does not match the release media.

Component	Validated version / identifier	Technician check
Technician manual	Version 1.0	This document
Hollyhock server release	_____	Match MANIFEST.txt
Public site release	_____	Match MANIFEST.txt
Database/schema level	_____	Match release notes
Caregiver Android app	_____	Match approved package/hash
Client & Family Android app	_____	Match approved package/hash

If any version is unknown or mismatched, stop and obtain the correct release-specific notes before proceeding.

Contents

Preflight gate and release/manual compatibility record

1. Installation map and server worksheet
2. Connect to the fresh Ubuntu server
3. Patch and harden the base operating system
4. Install Hollyhock server software
5. Create service accounts and filesystem layout
6. Transfer and verify the Developer Pack
7. Install and secure MySQL 8
8. Configure Hollyhock environment files
9. Install the immutable Hollyhock release
10. Run database validation and first migration
11. Install systemd services and timers
12. Configure DNS, Nginx, and HTTPS
13. Prove server health before user setup
14. First administrator login and password creation
15. Create offices, staff, caregiver, and family access
16. Create the first client and scheduled visit
17. Install the Android apps from the web application
18. Run the first CARE-GPS visit from Approach to Clock Out
19. Review the first visit from the office side
20. Backups, restore tests, upgrades, and rollback
21. Technician command reference

22. Regression and acceptance testing reference

23. Troubleshooting guide

24. External technical references and release handoff checklist

Installation Preflight Gate

Do not begin installation until every required item below is confirmed. Mark PASS or record the blocker. This is the point to stop a bad install before it becomes a recovery job.

Preflight item	PASS	Notes / blocker
Server meets the release CPU, memory, disk, and operating-system requirements.	[]	
Technician has root or working sudo access and a recovery/console path.	[]	
Customer controls the required DNS records and knows the intended hostnames.	[]	
Developer Pack release/version matches MANIFEST.txt and this manual compatibility record.	[]	
All release-media checksums pass before extraction or installation.	[]	
Backup destination and off-server copy location are identified.	[]	
Required public ports are known; database and Node service ports will remain private.	[]	
Customer/agency owner and technical contact are available for first-login/handoff decisions.	[]	
Mobile test device and safe acceptance-test location are available.	[]	

1. Installation Map and Server Worksheet

Do not start typing commands until you know the server name, public hostnames, server IP, and who controls DNS. Ten minutes of preparation saves an hour of fixing the wrong hostname later.

Item	Write the real value here
Server/VPS provider	_____
Server public IPv4	_____
Technician sudo user	_____
Public care website hostname	_____
Hollyhock application hostname	_____
Customer/agency name	_____
Primary office timezone	_____
Facilitator contact	_____
Release/version from MANIFEST.txt	_____

For a small pilot or staging-style installation, the current Hollyhock reference topology used 2 vCPU, 4 GiB RAM, an 80 GiB SSD, and 2 GiB swap. A larger agency may need more. A tiny 1 GiB server can run out of room quickly during builds, database work, or log growth.

Network rule. Only SSH (22), HTTP (80), and HTTPS (443) should be reachable from the public internet. MySQL and Hollyhock Node ports stay on loopback (127.0.0.1).

What the finished server looks like

```

Internet
|
+-- https://<PUBLIC_HOST> --> Nginx --> 127.0.0.1:4200 (public site)
|
+-- https://<APP_HOST>      --> Nginx --> 127.0.0.1:4100 (Hollyhock app/API/portal)
                                |
                                +--> MySQL 127.0.0.1:3306
                                +--> /var/lib/hollyhock state/uploads
                                +--> audit/evidence/logging

```

2. Connect to the Fresh Ubuntu Server

Most VPS providers give you an IP address and either a root password or an SSH key. The examples below use root for the first login because a brand-new server may not have a normal administrator account yet.

Connect from Linux, macOS, or Windows Terminal

```
ssh root@<SERVER_IP>
```

The first connection may ask whether you trust the host key. Confirm that the IP address is correct before accepting it. After login, confirm where you are:

```
whoami
hostname
hostname -I
lsb_release -a
```

Expected result. You should see root (or your authorized sudo user), the server hostname, the expected public IP, and Ubuntu 24.04 LTS.

Create a normal sudo technician account

Daily administration should not depend on a permanent root login. Create a normal technician account, give it sudo rights, and copy the working SSH key if the provider placed one under root.

```
adduser hollytech
usermod -aG sudo hollytech
install -d -m 700 -o hollytech -g hollytech /home/hollytech/.ssh
cp /root/.ssh/authorized_keys /home/hollytech/.ssh/authorized_keys
chown hollytech:hollytech /home/hollytech/.ssh/authorized_keys
chmod 600 /home/hollytech/.ssh/authorized_keys
```

Open a second terminal and prove the new account works before you close the root session:

```
ssh hollytech@<SERVER_IP>
```

If this fails, stop here and go to Troubleshooting T-01. Do not disable root access until the new account is proven.

SSH hardening closeout

After the normal sudo technician account is proven from a second terminal, decide the customer SSH policy before installation continues. Do not remove the only known-good access path.

Minimum closeout: confirm key-based login works, confirm the provider or console recovery path, restrict SSH at the provider firewall where practical, and document whether direct root login or password authentication remains permitted by customer policy. If either is disabled, open a new session and prove access again before closing the working session.

3. Patch and Harden the Base Operating System

Ubuntu LTS gives Hollyhock a predictable base. The first job is to bring the fresh server fully up to date, set the correct time, and make sure only the ports we need are open.

```
sudo apt update
sudo apt upgrade -y
sudo apt install -y ca-certificates curl gnupg unzip rsync git jq build-essential ufw fail2ban
```

If Ubuntu installs a new kernel, reboot now instead of discovering later that the server is waiting for one:

```
test -f /var/run/reboot-required && echo "Reboot required"
sudo reboot
```

Reconnect after the reboot. Check time synchronization:

```
timedatectl status
sudo timedatectl set-timezone <TIMEZONE>
```

Example timezone for eastern U.S. offices:

```
sudo timedatectl set-timezone America/New_York
```

Firewall

UFW is the simple host firewall used in this manual. It is not a replacement for the VPS provider firewall. If the provider supports a cloud firewall, restrict SSH to approved administrator addresses there as well.

```
sudo ufw allow OpenSSH
sudo ufw allow 'Nginx Full'
sudo ufw enable
sudo ufw status verbose
```

Do not open these ports publicly. 3306 (MySQL), 4100 (Hollyhock), and 4200 (public Node service) must remain private/loopback.

fail2ban verification

The base package installs fail2ban, but installation alone does not prove that SSH protection is active. If the Developer Pack includes a versioned fail2ban configuration, install that file and verify the named jail. Otherwise, record fail2ban as optional defense-in-depth and rely on SSH keys plus the VPS/provider firewall rather than assuming an unverified default jail is protecting the server.

```
sudo systemctl enable --now fail2ban
sudo systemctl status fail2ban --no-pager
sudo fail2ban-client status
```

If the release defines an sshd jail, verify it explicitly with the release-specific jail name. Do not enable a guessed configuration merely to make the command produce output.

4. Install Hollyhock Server Software

Hollyhock uses a small group of standard server tools. Each one has a clear job. If one is missing, the system becomes harder to operate or less secure.

Nginx: the public front door

Nginx accepts normal web traffic on ports 80 and 443. It does not own Hollyhock data. Its job is to terminate HTTPS, apply web security rules, and send each hostname to the correct local Node service.

```
sudo apt install -y nginx
sudo systemctl enable nginx
sudo systemctl status nginx --no-pager
```

MySQL 8: the core data store

MySQL stores the durable care-operations data: offices, users, clients, schedules, permissions, visit state, review state, and other relational records. It is bound to loopback so it is not exposed directly to the internet.

```
sudo apt install -y mysql-server mysql-client
sudo systemctl enable --now mysql
sudo systemctl status mysql --no-pager
sudo ss -ltnp | grep 3306
```

Expected result. MySQL should be active, and port 3306 should listen on 127.0.0.1 rather than 0.0.0.0.

Node.js 22 and npm 10: the Hollyhock runtime

Ubuntu 24.04 may offer an older Node.js package. The current Hollyhock baseline requires Node 22.22 or newer on the supported Node 22 line, with npm 10.x. This manual uses the NodeSource Node 22 repository.

```
curl -fsSL https://deb.nodesource.com/setup_22.x | sudo -E bash -
sudo apt install -y nodejs
node --version
npm --version
```

If Node reports a major version below 22, stop and see Troubleshooting T-03.

Certbot: browser-trusted HTTPS

Certbot obtains and renews Let's Encrypt certificates. Do not copy certificate private keys from another customer or another server.

```
sudo apt install -y certbot python3-certbot-nginx
certbot --version
```

5. Create Service Accounts and Filesystem Layout

A good Hollyhock server separates code, secrets, runtime state, logs, and backups. Code releases should be replaceable. Client data and uploads should not live inside a release directory.

```
sudo groupadd --system hhapp
sudo useradd --system --gid hhapp --home /nonexistent --shell /usr/sbin/nologin hhapp
sudo groupadd --system hhsite
sudo useradd --system --gid hhsite --home /nonexistent --shell /usr/sbin/nologin hhsite
sudo groupadd --system hhbackup
sudo useradd --system --gid hhbackup --home /nonexistent --shell /usr/sbin/nologin hhbackup
```

```
sudo mkdir -p /opt/hollyhock/releases /opt/hollyrandels/releases
sudo mkdir -p /var/lib/hollyhock/state /var/lib/hollyhock/uploads
sudo mkdir -p /var/lib/hollyrandels/state
sudo mkdir -p /var/log/hollyhock /var/log/hollyrandels
sudo mkdir -p /var/backups/hollyhock/database /var/backups/hollyhock/config
/var/backups/hollyhock/manifests
sudo mkdir -p /etc/hollyhock /etc/hollyrandels
sudo chmod 700 /etc/hollyhock /etc/hollyrandels
sudo chown -R hhapp:hhapp /var/lib/hollyhock /var/log/hollyhock
sudo chown -R hhsite:hhsite /var/lib/hollyrandels /var/log/hollyrandels
sudo chown -R hhbackup:hhbackup /var/backups/hollyhock
sudo chmod 700 /var/backups/hollyhock
```

Why this matters. A code rollback should not roll back or overwrite client data. Keeping state outside /opt/hollyhock/releases makes that separation possible.

6. Transfer and Verify the Developer Pack

Use SFTP or SCP. They are encrypted and use SSH. Do not install a plain FTP server just to move the Hollyhock package.

Copy the ISO from a workstation

```
scp HOLLYHOCK_SERVER_INSTALL.iso hollytech@<SERVER_IP>:/home/hollytech/
```

Or use interactive SFTP:

```
sftp hollytech@<SERVER_IP>
put HOLLYHOCK_SERVER_INSTALL.iso
bye
```

On the server, mount a downloaded ISO. A USB or physical disc may already be mounted by the operating system.

```
sudo mkdir -p /mnt/hollyhock
sudo mount -o loop /home/hollytech/HOLLYHOCK_SERVER_INSTALL.iso /mnt/hollyhock
ls -lah /mnt/hollyhock
```

A normal Developer Pack should contain a first-read file, documentation, install helpers, release archives, configuration templates, checksums, and the license material.

```
/README-FIRST.txt
/docs/
/install/
/config/
/releases/
/checksums/SHA256SUMS
/LICENSE/
```

Verify the package before installing it:

```
cd /mnt/hollyhock/checksums
sha256sum -c SHA256SUMS
```

If any checksum says FAILED. Stop. Do not install the damaged or altered package. Re-copy the media and verify it again.

7. Install and Secure MySQL 8

The MySQL root account is for server administration. Hollyhock should not run as database root. The recommended layout uses separate runtime, migration, and backup accounts so a compromised web process does not automatically receive schema-administration rights.

```
sudo mysql_secure_installation
```

Generate database-only secrets. These are machine secrets; they do not need to be easy to type manually.

```
openssl rand -hex 24
openssl rand -hex 24
openssl rand -hex 24
```

Store the three values in the customer password vault as Runtime DB, Migrator DB, and Backup DB. Then open MySQL with history disabled for this setup session:

```
MYSQL_HISTFILE=/dev/null sudo mysql
```

Run inside the MySQL prompt

```
CREATE DATABASE hollyhock CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci;

CREATE USER 'hollyhock_app'@'localhost' IDENTIFIED BY '<RUNTIME_DB_PASSWORD>';
GRANT SELECT, INSERT, UPDATE, DELETE ON hollyhock.* TO 'hollyhock_app'@'localhost';

CREATE USER 'hollyhock_migrator'@'localhost' IDENTIFIED BY '<MIGRATOR_DB_PASSWORD>';
GRANT ALL PRIVILEGES ON hollyhock.* TO 'hollyhock_migrator'@'localhost';

CREATE USER 'hollyhock_backup'@'localhost' IDENTIFIED BY '<BACKUP_DB_PASSWORD>';
GRANT SELECT, SHOW VIEW, TRIGGER, EVENT ON hollyhock.* TO 'hollyhock_backup'@'localhost';

FLUSH PRIVILEGES;
EXIT;
```

Release rule. If the Developer Pack includes a versioned SQL account/grant script, use the packaged script instead of retyping this example. The packaged release is authoritative for that release.

8. Configure Hollyhock Environment Files

Environment files contain the settings that change from customer to customer: hostnames, ports, database connection, feature switches, and secrets. Keep them outside the source tree and restrict them to root.

```
sudo nano /etc/hollyhock/api.env
```

Core application environment — example

```
NODE_ENV=production
HOST=127.0.0.1
PORT=4100
TRUST_PROXY=true
PUBLIC_ORIGIN=https://<APP_HOST>
DATABASE_URL=mysql://hollyhock_app:<URL_SAFE_RUNTIME_DB_PASSWORD>@127.0.0.1:3306/hollyhock
ENABLE_ADMIN_WEB=true
ENABLE_PUBLIC_INTAKE_API=true
ENABLE_CLIENT_PORTAL=true
ENABLE_CARE_GPS=true
ENABLE_DUTY_TIME=true
ENABLE_NATIVE_EVIDENCE=true
ENABLE_CLIENT_ANCHOR=false
ENABLE_DEMO_LAB=false
```

Now create the public site environment:

```
sudo nano /etc/hollyrandels/site.env
```

Public site environment — safe first-install state

```
NODE_ENV=production
HOST=127.0.0.1
PORT=4200
TRUST_PROXY=true
PUBLIC_ORIGIN=https://<PUBLIC_HOST>
HOLLYHOCK_API_BASE_URL=https://<APP_HOST>
ENABLE_SUBMISSIONS=false
```

Keep public submissions off until the application, signed intake handoff, database, and receipt checks have passed. Turning on a button before the receiving system is ready is not a successful deployment.

Create a separate migration environment that uses the migration-only database account. This file is used only during an approved schema change:

```
sudo nano /etc/hollyhock/migration.env
```

Migration environment — example

```
DATABASE_URL=mysql://hollyhock_migrator:<URL_SAFE_MIGRATOR_DB_PASSWORD>@127.0.0.1:3306/hollyhock
```

Create a MySQL client file for the backup account. This keeps the backup password off the mysqldump command line:

```
sudo nano /etc/hollyhock/backup.cnf
```

Backup client configuration — example

```
[client]
user=hollyhock_backup
password=<BACKUP_DB_PASSWORD>
host=127.0.0.1
```

```
sudo chown root:root /etc/hollyhock/api.env /etc/hollyhock/migration.env
/etc/hollyhock/backup.cnf /etc/hollyrandels/site.env
sudo chmod 600 /etc/hollyhock/api.env /etc/hollyhock/migration.env /etc/hollyhock/backup.cnf
/etc/hollyrandels/site.env
```

9. Install the Immutable Hollyhock Release

The recommended Developer Pack ships a versioned, prebuilt release. That makes the server install faster and avoids compiling the whole application on a small VPS. Source-build instructions belong in the developer appendix, not in the normal customer install path.

Read the release name from MANIFEST.txt. The commands below use <RELEASE_VERSION> as a placeholder.

```
sudo mkdir -p /opt/hollyhock/releases/<RELEASE_VERSION>
sudo tar -xzf /mnt/hollyhock/releases/hollyhock-<RELEASE_VERSION>.tar.gz -C
/opt/hollyhock/releases/<RELEASE_VERSION>

sudo mkdir -p /opt/hollyrandels/releases/<RELEASE_VERSION>
sudo tar -xzf /mnt/hollyhock/releases/hollyrandels-<RELEASE_VERSION>.tar.gz -C
/opt/hollyrandels/releases/<RELEASE_VERSION>

sudo ln -sf /opt/hollyhock/releases/<RELEASE_VERSION> /opt/hollyhock/current
sudo ln -sf /opt/hollyrandels/releases/<RELEASE_VERSION> /opt/hollyrandels/current
```

If the release does not ship production dependencies, install them from the included lockfiles before making the release read-only:

```
cd /opt/hollyhock/current
sudo npm ci --omit=dev

cd /opt/hollyrandels/current
sudo npm ci --omit=dev
```

Make the completed release tree non-writable to the runtime services:

```
sudo chown -R root:hhapp /opt/hollyhock/releases/<RELEASE_VERSION>
sudo chown -R root:hhsite /opt/hollyrandels/releases/<RELEASE_VERSION>
sudo find /opt/hollyhock/releases/<RELEASE_VERSION> -type d -exec chmod 0555 {} \;
sudo find /opt/hollyhock/releases/<RELEASE_VERSION> -type f -exec chmod 0444 {} \;
sudo find /opt/hollyrandels/releases/<RELEASE_VERSION> -type d -exec chmod 0555 {} \;
sudo find /opt/hollyrandels/releases/<RELEASE_VERSION> -type f -exec chmod 0444 {} \;
```

10. Run Database Validation and the First Migration

A fresh database still needs the Hollyhock schema. Do this before the application is exposed publicly. Do not manually force a migration number. Let the release migration tooling create the schema it expects.

Validate migration files first

```
cd /opt/hollyhock/current
npm run db:validate
```

The current Hollyhock codebase uses Drizzle migration tooling. For a fresh install, run the migration command shipped by the release. In the current reference code this has been exposed through the project database script. Run it with the migration environment, not the restricted runtime database account.

Current reference migration command for a fresh database

```
sudo bash -c 'set -a; . /etc/hollyhock/migration.env; set +a; cd /opt/hollyhock/current; npm run db:push'
```

Upgrade warning. A fresh install and an upgrade are not the same job. Do not use a fresh-install migration command to “repair” a populated server. See Section 20 before upgrading an existing database.

11. Install systemd Services and Timers

systemd keeps Hollyhock running after you log out and after the server reboots. The Developer Pack should supply service files so the technician does not have to invent them.

```
sudo cp /mnt/hollyhock/install/systemd/hollyhock.service /etc/systemd/system/
sudo cp /mnt/hollyhock/install/systemd/hollyrandels.service /etc/systemd/system/
sudo cp /mnt/hollyhock/install/systemd/hollyhock-backup.service /etc/systemd/system/
sudo cp /mnt/hollyhock/install/systemd/hollyhock-backup.timer /etc/systemd/system/

sudo systemctl daemon-reload
sudo systemctl enable --now hollyhock.service
sudo systemctl enable --now hollyrandels.service
sudo systemctl enable --now hollyhock-backup.timer
```

Check both services before touching Nginx:

```
sudo systemctl status hollyhock --no-pager
sudo systemctl status hollyrandels --no-pager
curl -fsS http://127.0.0.1:4100/health/live
curl -fsS http://127.0.0.1:4200/health/live
```

If you see “active (running)” and healthy responses. Continue to DNS and Nginx. If either service is failed, go to Troubleshooting T-06 before doing anything else.

12. Configure DNS, Nginx, and HTTPS

DNS tells the internet which server owns the customer hostnames. Nginx then decides which local Hollyhock service receives the request. Do not point DNS at the server until the local services are healthy.

Create A records at the customer DNS provider:

Type	Name	Value
A	<PUBLIC_SUBDOMAIN>	<SERVER_IP>
A	<APP_SUBDOMAIN>	<SERVER_IP>

Wait until public DNS returns the server IP:

```
dig +short <PUBLIC_HOST> A
dig +short <APP_HOST> A
```

Install the Nginx configuration supplied by the Developer Pack and replace the hostname placeholders. Keep the Node services on loopback.

```
sudo cp /mnt/hollyhock/install/nginx/hollyhock.conf /etc/nginx/sites-available/hollyhock
sudo ln -sfn /etc/nginx/sites-available/hollyhock /etc/nginx/sites-enabled/hollyhock
sudo rm -f /etc/nginx/sites-enabled/default
sudo nginx -t
sudo systemctl reload nginx
```

Obtain HTTPS certificates only after DNS points to this server:

```
sudo certbot --nginx -d <PUBLIC_HOST>
sudo certbot --nginx -d <APP_HOST>
sudo nginx -t
sudo systemctl reload nginx
sudo certbot renew --dry-run
```

Expected result. Both hostnames load over HTTPS with a browser-trusted certificate. HTTP redirects to HTTPS. The app hostname shows Hollyhock. The public hostname shows the customer public site.

13. Prove Server Health Before User Setup

Do not create users until the machine itself passes a small acceptance check. This gives you a clean dividing line between “server problem” and “account problem.”

```
curl -I https://<PUBLIC_HOST>/
curl -fsS https://<PUBLIC_HOST>/health/live
curl -I https://<APP_HOST>/app/login
curl -fsS https://<APP_HOST>/health/live
curl -fsS https://<APP_HOST>/health/ready
sudo nginx -t
sudo systemctl --failed
sudo ss -ltnp
```

You should see HTTPS 200/redirect behavior, healthy service endpoints, no failed Hollyhock units, Nginx syntax success, and only the expected public listeners.

14. First Administrator Login and Password Creation

The install media does not carry a permanent administrator password. The facilitator supplies a temporary account or secure invitation after the server is ready.

Open the application hostname in a browser and sign in with the temporary administrator email and password supplied by the facilitator. If the account is marked for first-login setup, Hollyhock will require the user to create a new password before normal work continues.

1. Open https://<APP_HOST>/app/login.
2. Enter the administrator email supplied by the facilitator.
3. Enter the temporary password.
4. Create a new private password when Hollyhock asks for it.
5. Sign out and sign in once more with the new password.

Do not put customer passwords in the manual. If a technician loses the temporary credential, use the authorized reset/invite flow. Do not create a permanent shared default password.

15. Create Offices, Staff, Caregiver, and Family Access

Hollyhock is multi-office. An organization administrator can be mapped to more than one office. An office user should see only the office they are allowed to use. A caregiver and a family user should not inherit staff administration permissions.

1. Create or confirm the customer organization.
2. Create the first office and set its timezone.
3. If the customer has another office, create it separately rather than pretending both locations are one office.
4. Create an administrator who can switch between authorized offices.
5. Create an office user for each office that needs its own local administration.
6. Create the first caregiver account and caregiver profile.
7. Create family/client portal access only after a client exists and an active client grant can be attached.

For every new user, use their email address as the login. Issue a temporary password or invitation and require a private password at first login. Do not reuse one password across five people just because it is a test day.

16. Create the First Client and Scheduled Visit

The first acceptance visit should use a demo/test client and a location that the tester can safely visit. Do not use a real client just to prove the installation.

1. Open Clients and create a Demo Test Client.
2. Add a service location with a real address the tester can physically visit, such as a grocery store used only for the test.
3. Confirm the map/GPS pin is on the correct location.
4. Open Schedule and create a visit for the test day and time.
5. Assign the caregiver account created in Section 15.
6. Publish/confirm the visit so it appears in the caregiver application.

The caregiver should be able to find the visit under My Day or Schedule before leaving for the test.

17. Install the Android Apps from the Web Application

Hollyhock can distribute its approved Android packages through the authenticated web application. This is useful before a store listing is available and for controlled customer deployments.

Verify the Android package before installation

The approved APK should be tied to the same release handoff as the server package. Before installing, compare the downloaded APK against the release checksum or approved package record. A filename alone is not proof that the file is the intended build.

```
sha256sum <APP_APK_FILE>
```

Compare the result with the checksum published in the verified release manifest or download page. For upgrades, also confirm package/signing continuity using the release notes. If the checksum or signer identity is unexpected, stop and obtain the approved package.

1. On the Android phone, open Chrome and go to https://<APP_HOST>.
2. Sign in with an account that has permission to download the mobile applications.
3. Open App Downloads / Android App Install.
4. Choose Hollyhock Caregiver or Hollyhock Client & Family.
5. Download the APK.

Chrome may warn that the file might be harmful because it is an APK downloaded outside Google Play. On a known Hollyhock server, continue with the approved file.

6. If Android asks, open Settings and allow Chrome to install apps from this source.
7. Return to the downloaded APK and continue installation.

Play Protect may say it has not seen an app from this developer before. If this is the approved Hollyhock build, open More details and choose Install anyway. Do not disable Play Protect globally.

8. Open the app and sign in with the caregiver or family mobile credentials supplied for that person.

What success looks like. The caregiver app opens to the field workflow with My Day, Schedule, Messages, and More. The family application shows only the family/client information authorized to that account.

18. Run the First CARE-GPS Visit: Approach to Clock Out

This is the point where the server install becomes a working care system. The goal is not to stare at the phone. The goal is to let the caregiver use Hollyhock naturally while the system records the visit workflow.

1. Open Hollyhock Caregiver before leaving.
2. Open the scheduled demo visit.
3. Choose Start Approach when travel to the service location begins.
4. Travel to the location normally. Keep the phone with the caregiver.
5. When Hollyhock shows that the caregiver has arrived or is ready for arrival confirmation, follow the on-screen arrival step.
6. Clock in only when the caregiver is actually ready to begin the visit.
7. Complete the visit normally. The app may run in the background; do not force-close it during the test.
8. Use tasks, observations, or messaging only if those parts are part of the acceptance test.
9. When the visit is finished, return to the app and clock out/end the visit.

CARE-GPS is intentionally not the same as a hidden automatic payroll clock. Location evidence supports the workflow, but the final visit state remains visible and reviewable. If location evidence is weak or a workflow is interrupted, Hollyhock should create a clear retry/exception path instead of silently inventing a clock event.

19. Review the First Visit from the Office Side

After clock-out, the office user should review the result before declaring the installation complete.

1. Open Schedule and confirm the visit shows the expected final state.
2. Open Hours and confirm the visit created the expected worked-time record.
3. Open GPS Review if the visit created a flagged or exception state.
4. Open the visit/reviewer timeline and confirm approach, arrival, clock-in, clock-out, and review evidence are present as expected.
5. Confirm the caregiver did not gain finance/admin permissions and the family account does not gain staff access.

First-install acceptance sentence. “First test completed successfully. Please look at analytics.” This is a useful handoff from the field tester to the administrator who will inspect the recorded visit.

20. Backups, Restore Tests, Upgrades, and Rollback

A backup that has never been restored is only a hopeful file. Hollyhock should make transaction-consistent database backups, hash them, keep them away from normal release directories, and periodically restore them into a separate test database.

Example transaction-consistent database backup

```
DATE=$(date -u +%Y%m%dT%H%M%SZ)
sudo mysqldump --defaults-extra-file=/etc/hollyhock/backup.cnf --single-transaction --routines
--events --triggers --no-tablespaces hollyhock | gzip >
/var/backups/hollyhock/database/hollyhock-$DATE.sql.gz

sudo sha256sum /var/backups/hollyhock/database/hollyhock-$DATE.sql.gz >
/var/backups/hollyhock/manifests/hollyhock-$DATE.sha256
```

Back up uploads, configuration, and release records

A database dump is not the whole installation. Preserve the files needed to rebuild the customer environment: approved configuration, uploads, manifests/checksums, and any release-specific state identified by the package.

```
DATE=$(date -u +%Y%m%dT%H%M%SZ)
sudo tar -czf /var/backups/hollyhock/config/hollyhock-config-$DATE.tar.gz /etc/hollyhock
/etc/hollyrandels
sudo tar -czf /var/backups/hollyhock/database/hollyhock-uploads-$DATE.tar.gz
/var/lib/hollyhock/uploads
sudo sha256sum /var/backups/hollyhock/config/hollyhock-config-$DATE.tar.gz
/var/backups/hollyhock/database/hollyhock-uploads-$DATE.tar.gz >
/var/backups/hollyhock/manifests/hollyhock-files-$DATE.sha256
```

These archives may contain secrets or client material. Keep them access-controlled, encrypt them when stored off-server, and copy at least one accepted backup away from the production server. If the release identifies additional writable state that must be transaction-consistent, follow the release-specific backup procedure instead of blindly copying a live directory.

For a restore test, create a separately named database. Never overwrite the active database just to see if the backup works.

```
sudo mysql -e "CREATE DATABASE hollyhock_restore_test CHARACTER SET utf8mb4 COLLATE
utf8mb4_0900_ai_ci;"
gunzip -c /var/backups/hollyhock/database/<BACKUP_FILE>.sql.gz | sudo mysql
hollyhock_restore_test
```

Upgrades require a compatibility check between code and database schema. A code rollback is safe only when the older code understands the current schema. If a migration changes authorization or schema meaning, do not blindly point the old application at the new database. Use the release recovery instructions, fix forward, or restore to a separately validated database.

Never do this. Do not “solve” a bad upgrade with an unplanned database downgrade, destructive migration reversal, or a copy command over a running database.

Rollback decision tree

Use the failure point to decide the recovery path. Do not improvise a database downgrade.

Failure point	Preferred action	Do not do this
Before schema migration	Return the current symlink/service to the last known-good release; re-run health checks.	Do not alter the database to match an install that never migrated.
After migration, before cutover	Use release-specific recovery instructions; fix forward when compatible, or restore a separately validated pre-upgrade database and matching code.	Do not point arbitrary old code at a changed schema.
After public cutover	Remove traffic from the failed release, preserve evidence/logs, and recover using the verified release notes and backup set.	Do not overwrite the active database or uploads while diagnosing.
Unknown state	Stop, preserve current files/logs/database backup, and escalate before further writes.	Do not keep retrying migrations until something appears to work.

21. Technician Command Reference

These commands are here so the technician does not have to remember the exact spelling at 2:00 a.m. Most are standard Linux commands. The Hollyhock-specific sections show what the result means for this system.

Find where you are and what is on disk

```
pwd
ls -lah
cd /path/to/folder
df -h
du -sh /var/lib/hollyhock /var/log/hollyhock /var/backups/hollyhock
find /opt/hollyhock -maxdepth 2 -type f | head -50
```

Find a command, package, or file

```
command -v nginx
which node
whereis mysql
apt search nginx
apt show nginx
dpkg -l | grep nginx
dpkg -L nginx
find /etc -name '*hollyhock*' 2>/dev/null
grep -R "PUBLIC_ORIGIN" /etc/hollyhock /etc/hollyrandels 2>/dev/null
```

Use `command --help` or `man command` when you need the built-in reference:

```
nginx -h
systemctl --help
man systemctl
man journalctl
man ss
```

Service and process checks

```
systemctl status hollyhock --no-pager
systemctl status hollyrandels --no-pager
systemctl --failed
journalctl -u hollyhock -n 100 --no-pager
journalctl -u hollyrandels -n 100 --no-pager
ps aux | grep node
ss -ltnp
```

Network, DNS, and HTTPS checks

```
hostname
hostname -I
ip addr
dig <PUBLIC_HOST> A
dig <APP_HOST> A
curl -I https://<PUBLIC_HOST>/
curl -I https://<APP_HOST>/app/login
curl -fsS https://<APP_HOST>/health/live
curl -fsS https://<APP_HOST>/health/ready
openssl s_client -connect <APP_HOST>:443 -servername <APP_HOST> </dev/null 2>/dev/null | openssl
x509 -noout -subject -issuer -dates
```

Nginx checks

```
sudo nginx -t
sudo systemctl status nginx --no-pager
sudo journalctl -u nginx -n 100 --no-pager
sudo systemctl reload nginx
```

A reload is preferred after a valid configuration change because it does not unnecessarily drop the running service.

MySQL checks

```
sudo systemctl status mysql --no-pager
sudo ss -ltnp | grep 3306
sudo mysql -e "SHOW DATABASES;"
sudo mysql -e "SELECT VERSION();"
sudo mysql -e "SHOW PROCESSLIST;"
```

Memory and disk checks

```
free -h
df -h
du -sh /var/log/* 2>/dev/null | sort -h | tail -20
```

```
journalctl --disk-usage  
uptime
```

Command result code

Immediately after a command, Linux stores its exit code. Zero means the command reported success. A non-zero value means it reported a problem.

```
echo $?
```

22. Regression and Acceptance Testing Reference

Tests are not only for developers. A technician can use them to answer a very practical question: is the application broken, or is the problem in DNS, Nginx, permissions, credentials, or data setup?

Test level	Question it answers	Typical use
Smoke test	Does the service start and answer?	After install, reboot, or service change.
Focused test	Does the specific feature work?	After changing auth, schedule, portal, GPS, or one UI route.
Regression test	Did we break something that used to work?	Before release or after a broad code change.
Acceptance test	Can a real user complete the workflow?	First install, external demo, release readiness.

Current Hollyhock project checks

Run these from the Hollyhock source directory when the Developer Pack includes source and development dependencies. Use the package lockfile. Do not run random global versions of test tools.

```
npm ci
npm run lint
npm run typecheck
npm test
npm run build
npm run db:validate
```

To see every script the installed release exposes:

```
npm run
```

For the public-site source tree, the current project has used its own check/build scripts. Run the scripts defined by that package rather than assuming the care app and public site are identical.

```
npm run check
npm run build
```

How to read an assertion failure

An assertion failure is useful. It tells you what the test expected and what the system actually returned. Do not change code just because a red line appears. First decide whether the application is wrong or the test setup is wrong.

```
Expected: 200
Received: 403
```

Meaning: the test expected access to succeed, but the server denied it. Check login state, role permissions, office mapping, CSRF, and route authorization before blaming Nginx.

```
Expected: "scheduled"
Received: "cancelled"
```

Meaning: the application ran, but the visit is in a different state than the test expected. Check the test record, schedule state, and previous workflow actions.

```
Expected rows: 1
Received rows: 0
```

Meaning: the expected database record was not created or could not be found. Check transaction rollback, data filters, office/client scope, and the database used by the test.

Do not create a test loop

A passing test is evidence. Do not rerun the entire suite every time you change one label. Run the smallest meaningful test first. Broaden only when the changed code can affect other workflows or when new evidence challenges an earlier result.

23. Troubleshooting Guide

When something fails, stop at the layer that failed. Do not rewrite Nginx because a password is wrong. Do not reset the database because DNS has not propagated. Work from the outside in: network, proxy, service, database, account, workflow.

T-01 — Cannot SSH to the server

Check the server IP, cloud firewall, and port 22. From your workstation run:

```
ssh -vvv hollytech@<SERVER_IP>
```

On a provider console, check:

```
sudo systemctl status ssh --no-pager
sudo ss -ltnp | grep :22
sudo ufw status verbose
```

If SSH is not listening, repair SSH before touching Hollyhock.

T-02 — apt install or apt update fails

Check DNS and network access:

```
ping -c 2 1.1.1.1
getent hosts archive.ubuntu.com
sudo apt update
```

If IP connectivity works but names do not resolve, fix DNS. If the package database is locked, wait for unattended-upgrades to finish rather than killing dpkg blindly.

T-03 — Node.js version is wrong

Run:

```
node --version
npm --version
apt-cache policy nodejs
```

Hollyhock currently expects the supported Node 22 line. If Ubuntu installed Node 18, remove the old package and install the approved NodeSource 22 package.

T-04 — MySQL will not start

Run:

```
sudo systemctl status mysql --no-pager
sudo journalctl -u mysql -n 150 --no-pager
sudo ss -ltnp | grep 3306
df -h
```

Common causes are a full disk, bad MySQL configuration, permission problems, or an incomplete package upgrade. Do not delete /var/lib/mysql to “start clean.”

T-05 — Database migration fails

Stop the deployment. Record the migration error and current release version. Run the release migration validator and confirm you are pointing at the intended database. For a populated installation, do not rerun a fresh-install migration blindly. Preserve a backup and use the upgrade/recovery procedure for that release.

T-06 — Hollyhock service will not start

Run:

```
sudo systemctl status hollyhock --no-pager
sudo journalctl -u hollyhock -n 150 --no-pager
command -v node
ls -lah /opt/hollyhock/current
sudo test -r /etc/hollyhock/api.env && echo "api.env readable by root"
```

Look for a missing environment value, bad database connection, wrong release path, unsupported Node version, or permission error.

T-07 — Browser shows 502 Bad Gateway

A 502 usually means Nginx is alive but the local application is not answering. Run:

```
curl -v http://127.0.0.1:4100/health/live
sudo systemctl status hollyhock --no-pager
sudo nginx -t
sudo journalctl -u nginx -n 100 --no-pager
```

Fix the local service first. Do not change DNS because of a 502.

T-08 — HTTPS certificate fails

Confirm DNS first:

```
dig +short <APP_HOST> A
dig +short <PUBLIC_HOST> A
```

Then run:

```
sudo nginx -t
sudo certbot certificates
sudo certbot renew --dry-run
```

Let's Encrypt cannot validate a hostname that still points to another server.

T-09 — User cannot log in

Confirm the user is active, is using the correct email, and is on the correct login route. If the user forgot the password, use the reset flow. Do not read password hashes or create a shared master password. If login succeeds but the wrong office appears, inspect office mapping and role assignment.

T-10 — Android app will not install

Confirm the APK came from the authorized Hollyhock download page. Allow Chrome to install from this source if Android asks. If Play Protect says the developer is uncommon, open More details and use Install anyway for the approved build. Do not disable Play Protect globally. If updating, confirm package/signing continuity.

T-11 — Caregiver cannot see the scheduled visit

Check that the visit is assigned to that caregiver, published/scheduled, inside the app's date window, and belongs to the caregiver's authorized office. Check Schedule in the web app before changing the Android installation.

T-12 — Approach/GPS/arrival does not behave as expected

Check location permission, device location services, time, visit schedule, service-location pin, and app authorization. Do not manufacture a clock event from a weak GPS result. Use the retry/exception path and review the evidence timeline.

T-13 — Disk is full or nearly full

Run:

```
df -h
du -sh /var/log/* 2>/dev/null | sort -h | tail -20
du -sh /var/backups/hollyhock/* 2>/dev/null
journalctl --disk-usage
```

Clean only known logs, old generated artifacts, and backups according to the retention policy. Never delete /var/lib/mysql, active uploads, or the current release because the disk is full.

T-14 — Backup restore test fails

Do not overwrite the active database. Restore again into a new separately named test database, verify the backup checksum, inspect mysqldump/mysql errors, and confirm the backup account had the privileges required by the release. A failed restore means the backup process is not accepted yet.

24. External Technical References and Release Handoff Checklist

The Developer Pack must be usable without browsing the internet after download. These references are still useful for a technician who wants the full vendor documentation.

Tool	Official reference
Ubuntu Server	https://ubuntu.com/server/docs
Node.js	https://nodejs.org/
NodeSource distributions	https://github.com/nodesource/distributions
MySQL 8.0 Reference Manual	https://dev.mysql.com/doc/refman/8.0/en/
Nginx documentation	https://nginx.org/en/docs/
systemd manuals	https://www.freedesktop.org/software/systemd/man/
Let's Encrypt	https://letsencrypt.org/docs/
Certbot	https://eff-certbot.readthedocs.io/
Git	https://git-scm.com/docs

Final technician handoff checklist

- Ubuntu 24.04 LTS is patched and time is correct.
- SSH works through the normal sudo technician account.
- Only approved public ports are open.
- Node.js and npm meet the release requirements.
- MySQL is active and loopback-only.
- Hollyhock runtime, migration, and backup database accounts are separate.
- Application and public-site releases are installed from verified media.
- Environment files are outside the source tree and mode 0600.
- systemd services start at boot and health checks pass.
- DNS resolves to the intended server.
- Nginx configuration passes `nginx -t`.
- HTTPS is valid and Certbot renewal dry-run passes.
- Administrator completed first-login password creation.
- Office mapping and role boundaries were checked.
- A demo client and service location were created.
- A caregiver installed the Android application from the web app.
- Approach, arrival, clock-in, visit, and clock-out were completed.
- The office reviewed the completed visit and evidence.
- A database backup was created, hashed, and restore-tested separately.
- The customer received the license, release manifest, and this manual.

Customer Handoff Bundle

Before the technician signs off, confirm that the customer or authorized administrator received the operational materials required to run and recover the installation without depending on the installer's memory.

Handoff item	Delivered
Coal Road software license and product/release identifier	[]
Verified release media or approved download location	[]
MANIFEST.txt and SHA256SUMS / verification record	[]
Technician manual and administrator guide	[]
Customer server worksheet and installed hostnames	[]
Credential-vault handoff for administrator and machine secrets (not printed passwords)	[]
DNS ownership/contact and certificate-renewal responsibility	[]
Backup locations, retention notes, and last successful restore-test record	[]
Android package/version/checksum information used for acceptance	[]
Known limitations, deferred integrations, and release-specific follow-up items	[]

Technician sign-off

Field	Value
Customer / Agency	_____
Server hostname	_____
Public hostname	_____
Application hostname	_____
Installed Hollyhock release	_____
Installation date	_____
Technician	_____
First caregiver acceptance visit	PASS / FAIL
Backup restore test	PASS / FAIL

End state. The installation is not finished because the login page looks good. It is finished when the server is healthy, permissions are correct, a caregiver can complete the first authorized visit, the office can review it, and a tested backup exists.

Developer Pack Distribution Notes

This technician manual is designed to ship with the Hollyhock Developer Pack on downloadable media, USB, mounted ISO, or physical optical media. The same package layout should be used on every medium so the instructions do not change just because the customer bought a disc instead of a download.

A physical edition can be a fun and useful artifact, but the install must never depend on an optical drive. The ISO should be mountable directly, burnable to DVD if desired, and copyable to USB without changing the contents.

Support, hosting, implementation, migration, and integration are separate services unless the customer agreement says otherwise. The software package should remain usable by a competent technician with the release files, license, and documentation in hand.

Version 1.0 release rule: before this manual ships with a Developer Pack, validate every Hollyhock-specific command, service filename, environment key, health route, migration command, APK checksum process, and package path against the exact distributable release. Generic Linux guidance may remain stable; the release-specific manifest controls when the two disagree.