From GitHub to Forgejo

Forgejo on Umbrel with Cloudflare Tunnel: Complete Playbook

Cody Brunner · codybrunner.com · May 2026

A complete reference for setting up a self-hosted Forgejo instance on an Umbrel home server, exposing it via Cloudflare Tunnels, and securing it with Cloudflare Zero Trust Access.

Sections:

> *WARNING This guide assumes an Umbrel home server. For manual Docker Compose installation, see the official Forgejo docs: https://forgejo.org/docs/latest/admin/installation/*

1. Environment Reference

Hardware & Network

Component	Detail
Device	Umbrel home server
LAN IP	192.168.x.x
Router	your home router
Domain	example.com (managed on Cloudflare)

Forgejo Stack

Component	Detail
Forgejo image	codeberg.org/forgejo/forgejo:15.0.2-rootless
Database	PostgreSQL 17.3
Web port (internal)	8101
SSH port (internal)	2223
Forgejo container	forgejo_server_1
Database container	forgejo_db_1
Umbrel app proxy	forgejo_app_proxy_1
Docker network	umbrel_main_network

Cloudflare Tunnel

Component	Detail
Tunnel name	your-tunnel-name
Tunnel ID	
Connector container	cloudflared_connector_1
Other containers	cloudflared_web_1, cloudflared_app_proxy_1

Subdomains

Subdomain	Purpose

forgejo.example.com	Forgejo web UI
git.example.com	SSH access for Git push/pull/clone

Key File Paths (on host)

Item	Path
docker-compose.yml	/home/umbrel/umbrel/app-data/forgejo/docker-compose.yml
app.ini	/home/umbrel/umbrel/app-data/forgejo/data/forgejo/data/custom/conf/app.ini
Git repositories	/home/umbrel/umbrel/app-data/forgejo/data/forgejo/data/git/
PostgreSQL data	/home/umbrel/umbrel/app-data/forgejo/data/db/
SSH host keys	/home/umbrel/umbrel/app-data/forgejo/data/forgejo/data/ssh/
Exports (env vars)	/home/umbrel/umbrel/app-data/forgejo/exports.sh
Config mount (/etc/gitea)	Empty — unused; app.ini lives in the data volume

2. How Forgejo Configuration Works on Umbrel

Forgejo uses two layers of configuration. Understanding precedence is critical:

- **1. Environment variables** in docker-compose.yml — highest priority, override app.ini
- **2. app.ini** — applies for any setting NOT defined as an env var

The environment variables already defined in docker-compose.yml control: `DOMAIN`, `HTTP_PORT`, `SSH_DOMAIN`, `SSH_PORT`, `SSH_LISTEN_PORT`, `START_SSH_SERVER`, `INSTALL_LOCK`, and all database settings. You cannot override these via app.ini — the env vars always win.

Settings not covered by env vars (registration policy, session security, etc.) go in app.ini and survive Umbrel app updates since app.ini lives in the persistent data volume.

The Forgejo image is rootless, which means:

- The built-in SSH server is the only option (no system OpenSSH)
- The `gitea/` path naming inside containers is inherited from Gitea and is expected/correct
- app.ini lives at `/var/lib/gitea/custom/conf/app.ini` inside the container
- The `/etc/gitea` config mount exists but is empty and unused

3. Forgejo Configuration

3.1 docker-compose.yml Changes

File: `/home/umbrel/umbrel/app-data/forgejo/docker-compose.yml`

Three changes to existing env vars, plus one new env var added to the **server** service **environment:** block:

```
# Changed from ${APP_DOMAIN} to hardcoded public hostnames:
FORGEJO__server__DOMAIN: forgejo.example.com
FORGEJO__server__SSH_DOMAIN: git.example.com

# Changed from ${APP_FORGEJO_SSH_PORT} to 22 (display port only):
FORGEJO__server__SSH_PORT: 22

# Added new — was not present in the default compose at all:
FORGEJO__server__ROOT_URL: https://forgejo.example.com/

# Unchanged — actual port the server binds to inside the container:
FORGEJO__server__SSH_LISTEN_PORT: ${APP_FORGEJO_SSH_PORT}
```

Why each change matters:

- `DOMAIN` — controls the hostname shown in the Forgejo web UI. The default `${APP_DOMAIN}` resolves to umbrel.local, which breaks external URLs.
- `SSH_DOMAIN` — controls the hostname in SSH clone URLs. Same problem as above.
- `SSH_PORT` set to 22 — produces clean clone URLs (`git@git.example.com:user/repo.git`). This is the port *displayed* in the UI, not the port the server listens on.
- `ROOT_URL` (new) — controls all generated links including emails, webhooks, and OAuth redirects. Without it, Forgejo builds HTTP links pointing to port 8101 instead of your public HTTPS domain.
- `SSH_LISTEN_PORT` stays as `${APP_FORGEJO_SSH_PORT}` (2223) — the actual port the built-in SSH server binds to inside the container.

TIP Before editing, always save a backup of the original file:

```
cp /home/umbrel/umbrel/app-data/forgejo/docker-compose.yml \
   /home/umbrel/umbrel/app-data/forgejo/docker-compose.yml.bak
```

WARNING CRITICAL: This file gets overwritten on every Umbrel app update. All changes must be re-applied after each Forgejo update. This is confirmed behavior — Umbrel pulls a fresh compose file from the app store on every update with no built-in merge or override mechanism. Keep your backup and a diff of your changes. See Section 11 for the post-update recovery checklist.

3.2 The Full docker-compose.yml (with changes applied)

```
version: '3.7'
services:
```

```
app_proxy:
  environment:
    APP_HOST: forgejo_server_1
    APP_PORT: 8101
    PROXY_AUTH_ADD: 'false'
  container_name: forgejo_app_proxy_1
server:
  image: >-
    codeberg.org/forgejo/forgejo:15.0.2-rootless@sha256:cc2d74fb4c30385a8ee34de8c8f83344f7316cec70bd2cc
  user: '1000:1000'
  restart: on-failure
  ports:
    - ${APP_FORGEJO_SSH_PORT}:${APP_FORGEJO_SSH_PORT}
  volumes:
    - ${APP_DATA_DIR}/data/forgejo/data:/var/lib/gitea
    - ${APP_DATA_DIR}/data/forgejo/config:/etc/gitea
  environment:
    FORGEJO__security__INSTALL_LOCK: 'true'
    FORGEJO__server__DOMAIN: forgejo.example.com       # changed: was ${APP_DOMAIN}
    FORGEJO__server__HTTP_PORT: 8101
    FORGEJO__server__SSH_DOMAIN: git.example.com       # changed: was ${APP_DOMAIN}
    FORGEJO__server__SSH_PORT: 22                      # changed: was ${APP_FORGEJO_SSH_PORT}
    FORGEJO__server__SSH_LISTEN_PORT: ${APP_FORGEJO_SSH_PORT}
    FORGEJO__server__START_SSH_SERVER: 'true'
    FORGEJO__server__ROOT_URL: https://forgejo.example.com/  # added: not in default
    FORGEJO__database__DB_TYPE: postgres
    FORGEJO__database__HOST: forgejo_db_1:5432
    FORGEJO__database__NAME: forgejo
    FORGEJO__database__USER: forgejo
    FORGEJO__database__PASSWD: forgejo
  depends_on:
    db:
      condition: service_healthy
  container_name: forgejo_server_1
db:
  image: >-
    postgres:17.3@sha256:0321e2252ebfeecb8bc1a899755084d29bce872953e1a5a3e25ec0860b739098
  restart: on-failure
  user: '1000:1000'
  environment:
    POSTGRES_USER: forgejo
    POSTGRES_PASSWORD: forgejo
    POSTGRES_DB: forgejo
  volumes:
    - ${APP_DATA_DIR}/data/db:/var/lib/postgresql/data
  healthcheck:
    test:
      - CMD-SHELL
      - pg_isready -U forgejo
    interval: 5s
```

```
      timeout: 5s
      retries: 5
    container_name: forgejo_db_1
```

3.3 app.ini Changes

File: `/home/umbrel/umbrel/app-data/forgejo/data/forgejo/data/custom/conf/app.ini`

> *TIP Before editing, save a backup:*

```
cp /home/umbrel/umbrel/app-data/forgejo/data/forgejo/data/custom/conf/app.ini \
   /home/umbrel/umbrel/app-data/forgejo/data/forgejo/data/custom/conf/app.ini.bak
```

Unlike docker-compose.yml, app.ini lives in the persistent data volume and survives Umbrel app updates.

Three changes made to the existing file:

- **[server]** — set `ROOT_URL = https://forgejo.example.com/` (was blank). Optional given the docker-compose env var, but makes the file self-documenting.
- **[service]** — changed `DISABLE_REGISTRATION` from false to true
- **[service]** — changed `REQUIRE_SIGNIN_VIEW` from false to true

These settings go in app.ini (not docker-compose.yml) because they are not already defined as env vars, or because they complement the env var with an explicit value for clarity. They survive Umbrel app updates since app.ini lives in the persistent data volume.

3.4 The Full app.ini (with changes applied)

```
APP_NAME = Your Forge Name
RUN_USER = git
RUN_MODE = prod
WORK_PATH = /var/lib/gitea

[repository]
ROOT = /var/lib/gitea/git/repositories

[repository.local]
LOCAL_COPY_PATH = /tmp/gitea/local-repo

[repository.upload]
TEMP_PATH = /tmp/gitea/uploads

[server]
APP_DATA_PATH = /var/lib/gitea
SSH_DOMAIN = umbrel.local
HTTP_PORT = 8101
ROOT_URL = https://forgejo.example.com/   ; changed: was blank
DISABLE_SSH = false
; In rootless gitea container only internal ssh server is supported
```

```
START_SSH_SERVER = true
SSH_PORT = 2223
SSH_LISTEN_PORT = 2223
BUILTIN_SSH_SERVER_USER = git
LFS_START_SERVER =
DOMAIN = umbrel.local

[database]
PATH = /var/lib/gitea/data/gitea.db
DB_TYPE = postgres
HOST = forgejo_db_1:5432
NAME = forgejo
USER = forgejo
PASSWD = forgejo

[session]
PROVIDER_CONFIG = /var/lib/gitea/data/sessions

[picture]
AVATAR_UPLOAD_PATH = /var/lib/gitea/data/avatars
REPOSITORY_AVATAR_UPLOAD_PATH = /var/lib/gitea/data/repo-avatars

[attachment]
PATH = /var/lib/gitea/data/attachments

[log]
ROOT_PATH = /var/lib/gitea/data/log

[security]
INSTALL_LOCK = true
SECRET_KEY =
REVERSE_PROXY_LIMIT = 1
REVERSE_PROXY_TRUSTED_PROXIES = *
INTERNAL_TOKEN = <your-internal-token>

[service]
DISABLE_REGISTRATION = true            ; changed: was false
REQUIRE_SIGNIN_VIEW = true             ; changed: was false

[lfs]
PATH = /var/lib/gitea/git/lfs

[oauth2]
JWT_SECRET = <your-jwt-secret>
```

TIP The [server] section values for DOMAIN, SSH_DOMAIN, SSH_PORT, and SSH_LISTEN_PORT still show their original umbrel.local defaults. This is expected — the environment variables in docker-compose.yml override them at runtime. Only ROOT_URL is set here explicitly as a complement to the env var.

3.5 Admin Account

The first user registered via the Forgejo web UI is automatically granted admin privileges. No CLI command needed. Verify admin status:

```
sudo docker exec forgejo_server_1 forgejo admin user list
```

Or check for **Site Administration** in the avatar dropdown menu in the web UI.

3.6 Restarting Forgejo

Umbrel injects required environment variables into containers. Running `docker compose` directly from the CLI fails with blank variable warnings. Three options:

Option A — Umbrel dashboard (safest):

Restart the Forgejo app from the Umbrel web UI.

Option B — Source variables first:

```
export APP_DATA_DIR="/home/umbrel/umbrel/app-data/forgejo"
source /home/umbrel/umbrel/app-data/forgejo/exports.sh
cd /home/umbrel/umbrel/app-data/forgejo
sudo -E docker compose down
sudo -E docker compose up -d
```

The `-E` flag preserves environment variables through sudo.

Option C — Umbrel scripts (if available):

```
sudo /home/umbrel/umbrel/scripts/app restart forgejo
```

3.7 SSH Clone URL Format

The built-in SSH server causes Forgejo to generate `ssh://` style URLs:

```
ssh://git@git.example.com/your-username/dotfiles.git
```

This is normal and functionally identical to the SCP-style shorthand (`git@git.example.com:user/repo.git`). The ~/.ssh/config ProxyCommand works with either format.

4. Cloudflare Tunnel Setup

4.1 Create the Tunnel

Step 1

- Go to one.dash.cloudflare.com → Networks → Tunnels

Step 2

- Click Create a tunnel → select Cloudflared

Step 3

- Name: your-tunnel-name

Step 4

- Select Docker as operating system to isolate the token easily

Step 5

- Copy the connector token (the eyJhIjoi... string after --token)

4.2 Connect the Umbrel App

Step 1

- Open the Cloudflare Tunnel app in the Umbrel dashboard

Step 2

- Paste the token into the configuration field

Step 3

- Save and restart

Step 4

- Verify tunnel shows as Healthy in the Cloudflare dashboard

4.3 Add Two Public Hostname Routes

In Cloudflare Zero Trust dashboard: Networks → Tunnels → your-tunnel-name → Routes tab → + Add route → Published application.

Route 1: Forgejo Web UI

Field	Value
Subdomain	forgejo
Domain	example.com
Path	empty
Service URL	http://192.168.x.x:8101

Uses the Umbrel LAN IP directly. Bypasses Umbrel's app proxy intentionally — Forgejo has its own auth layer, plus Cloudflare Access on top.

Route 2: Git SSH

Field	Value
Subdomain	git
Domain	example.com
Path	empty
Service URL	tcp://192.168.x.x:2223

TCP protocol, LAN IP, port 2223 — matches APP_FORGEJO_SSH_PORT from Umbrel's exports.sh.

4.4 Why the Routes Work

Both routes use the Umbrel LAN IP rather than container names. The Cloudflare connector resolves the LAN IP through the tunnel's outbound connection — no inbound ports need to be opened on the router. Verify connectivity from the Umbrel:

```
curl -s -o /dev/null -w "%{http_code}\n" http://192.168.x.x:8101
# Expected: 200 or 302
```

4.5 DNS Records (Automatic)

When you add a public hostname, Cloudflare automatically creates CNAME records pointing each subdomain to `<tunnel-UUID>.cfargotunnel.com`. If auto-creation fails, manually add in DNS → Records:

Type	Name	Target	Proxy
CNAME	forgejo	.cfargotunnel.com	Proxied
CNAME	git	.cfargotunnel.com	Proxied

4.6 Local Network DNS Gotcha

When on the same LAN as the Umbrel, your router may struggle with hairpin NAT — resolving forgejo.example.com through Cloudflare back to your local machine. If subdomains do not resolve at home, add local DNS overrides on the router pointing to 192.168.x.x, or access Forgejo directly via http://192.168.x.x:8101 on the local network.

5. Cloudflare Zero Trust Access

5.1 Identity Provider

Enable One-time PIN in Zero Trust dashboard: Settings → Authentication → Login methods. Sends a code to your email with zero external configuration.

5.2 Access Policy

Create a reusable policy named **Allow Only Me**. At minimum, require Email. Add any additional factors that fit your situation:

- **Email** (required) — restricted to your email address
- **Country** — restrict to one or more countries
- **IP ranges** — lock to your home or VPN IP range
- **Authentication method** — require a specific IdP or MFA method

Multiple rules in the same policy are evaluated with AND logic — all must pass. Email alone is reasonable for a personal instance. Adding country or IP makes it meaningfully harder for anyone who obtains your OTP code to do anything useful with it.

5.3 Access Applications

Create two applications, one per exposed subdomain. No wildcard catch-all — other subdomains on example.com may host public apps that should not require authentication.

Forgejo-SSH

Field	Value
Application name	Forgejo-SSH
Session Duration	24 hours
Public hostname subdomain	git
Public hostname domain	example.com
Browser rendering	Disabled
Allow automatic Cloudflared authentication	On
Access policies	Allow Only Me
Apply instant authentication	On

Advanced settings:

Field	Value

HTTP Only	Off
Enable Binding Cookie	Off
Same Site Attribute	empty
Return 401 Response	Off

TIP "Enable Binding Cookie" explicitly says "Do not use for non-HTTP applications that rely on protocols like SSH and RDP" — correctly left off. "Allow automatic Cloudflared authentication" is on so cloudflared on the dev machine can use cached tokens without forcing a browser popup on every Git operation.

Forgejo-Web

Field	Value
Application name	Forgejo-Web
Session Duration	24 hours
Public hostname subdomain	forgejo
Public hostname domain	example.com
Browser rendering	Disabled
Allow automatic Cloudflared authentication	Off
Access policies	Allow Only Me
Apply instant authentication	On

Advanced settings:

Field	Value
HTTP Only	On
Enable Binding Cookie	On
Same Site Attribute	Lax
Return 401 Response	On

TIP Return 401 Response is required for Git HTTPS operations — without it, git clone gets redirected to the login page instead of receiving a clean 401 that triggers service token auth.

5.4 Why No Wildcard Catch-All

A wildcard *.example.com Access application would put an authentication gate in front of every subdomain, including any public apps hosted elsewhere on the same domain. Cloudflare Access applies at the edge based on hostname regardless of whether traffic routes through a tunnel or to an external origin. Two explicit applications cover everything exposed through the Forgejo tunnel.

6. Forgejo Runner (CI/CD)

The Forgejo runner is a manual install — there is no Umbrel App Store shortcut. This section covers the UmbrelOS-specific setup, which deviates from the official docs in a few deliberate ways. Every deviation is called out.

6.1 Architecture

Component	Value
Runner image	data.forgejo.org/forgejo/runner:12
Container name	forgejo_runner_1
Host directory	/home/umbrel/umbrel/app-data/forgejo-runner/
Runner UID:GID	1001:1001
Executor strategy	Host Docker socket mount
Forgejo internal URL	http://forgejo_server_1:8101
Docker network	umbrel_main_network (external)

Default labels:

Label	Image
node	node:24-bookworm
go	ghcr.io/catthehacker/ubuntu:act-22.04
docker	ghcr.io/catthehacker/ubuntu:act-22.04

> *TIP Pin the runner image to a major tag (:12) rather than a specific patch so security patches are picked up automatically on docker compose pull. When Forgejo jumps a major version, verify runner compatibility before bumping.*

6.2 Deviations from the Official Docs

Topic	Official docs	This guide	Reason
Executor	Docker-in-Docker sidecar	Host Docker socket + automount	DinD containers can't resolve forgejo_server_1 without complex network plumbing
Runner network	Default Compose network	umbrel_main_network (external)	Required to register against http://forgejo_server_1:8101 over Docker's internal DNS

container.network	Empty (isolated per job)	umbrel_main_network	Job containers need to reach forgejo_server_1 for actions/checkout
Install location	Unspecified	/home/umbrel/umbrel/app-data/forgejo-runner/	Critical: UmbrelOS only persists files under app-data/ across OS updates
Forgejo URL at registration	Public hostname	http://forgejo_server_1:8101	Bypasses Cloudflare Access and uses Docker network directly

> *WARNING SECURITY TRADE-OFF: With socket mount, a compromised workflow can manipulate any container on the Umbrel. This is acceptable only when you are the sole author of workflows and do not enable Actions on repos where untrusted contributors can push.*

6.3 UmbrelOS Persistence Model

UmbrelOS only guarantees files under /home/umbrel/umbrel/app-data/ survive OS updates. Install the runner there. Anywhere else is unsafe.

Path	Survives OS updates?
/home/umbrel/umbrel/app-data/	Yes
/home/umbrel/ (outside umbrel/)	No — gets wiped
/etc/, /usr/, /opt/	No — reset on update
Custom systemd units, /etc/group edits	No

6.4 Pre-flight Verification

SSH into Umbrel and run these before starting:

```
# 1. Confirm Forgejo container name and network attachment
sudo docker inspect forgejo_server_1 --format \
  '{{range $k, $v := .NetworkSettings.Networks}}{{$k}}{{"\n"}}{{end}}'
# Expected: umbrel_main_network

# 2. Confirm Forgejo is reachable on port 8101
sudo docker run --rm --network umbrel_main_network curlimages/curl:latest \
  -s -o /dev/null -w "%{http_code}\n" http://forgejo_server_1:8101/
# Expected: 200 or 3xx

# 3. Find the docker group GID on the host
stat -c '%g' /var/run/docker.sock
# Typically 987 or 999 on UmbrelOS — verify, don't assume.

# 4. Confirm Forgejo Actions is enabled globally
sudo grep -A2 '^\[actions\]' \
  /home/umbrel/umbrel/app-data/forgejo/data/forgejo/data/custom/conf/app.ini \
  || echo "No [actions] section — using defaults (enabled)."
```

6.5 Installation

Step 1: Create the runner directory

```
sudo mkdir -p /home/umbrel/umbrel/app-data/forgejo-runner/data/.cache
sudo chown -R 1001:1001 /home/umbrel/umbrel/app-data/forgejo-runner/data
sudo chmod -R 775 /home/umbrel/umbrel/app-data/forgejo-runner/data
sudo chmod g+s /home/umbrel/umbrel/app-data/forgejo-runner/data/.cache
```

Step 2: Generate and edit runner-config.yml

```
cd /home/umbrel/umbrel/app-data/forgejo-runner
sudo docker run --rm --user 1001:1001 \
  --entrypoint forgejo-runner \
  data.forgejo.org/forgejo/runner:12 \
  generate-config > /tmp/runner-config.yml
sudo mv /tmp/runner-config.yml \
  /home/umbrel/umbrel/app-data/forgejo-runner/data/runner-config.yml
sudo chown 1001:1001 \
  /home/umbrel/umbrel/app-data/forgejo-runner/data/runner-config.yml
sudo chmod 664 \
  /home/umbrel/umbrel/app-data/forgejo-runner/data/runner-config.yml
```

Replace the generated file contents with this UmbrelOS-tuned config:

```
log:
  level: info
runner:
  file: .runner
  capacity: 3
```

```
  envs: {}
  labels: []
  timeout: 3h
  shutdown_timeout: 3h
  insecure: false
  fetch_timeout: 5s
  fetch_interval: 2s
  report_interval: 1s
cache:
  enabled: true
  dir: ""
  host: ""
  port: 0
  external_server: ""
container:
  # UMBREL-SPECIFIC: attach job containers to umbrel_main_network
  # so they can reach forgejo_server_1:8101 for checkout.
  network: umbrel_main_network
  privileged: false
  options: ""
  workdir_parent: ""
  valid_volumes: []
  # Auto-detect and mount the host Docker socket into job containers.
  docker_host: automount
  force_pull: false
  force_rebuild: false
host:
  workdir_parent: ""
```

Step 3: Create docker-compose.yml

Replace **987** with the GID from pre-flight Step 3 (`stat -c '%g' /var/run/docker.sock`):

```
services:
  runner:
    image: data.forgejo.org/forgejo/runner:12
    container_name: forgejo_runner_1
    restart: unless-stopped
    user: "1001:1001"
    group_add:
      - "987"        # REPLACE with your docker.sock GID
    networks:
      - umbrel_main_network
    volumes:
      - ./data:/data
      - /var/run/docker.sock:/var/run/docker.sock
    command: forgejo-runner daemon --config /data/runner-config.yml
networks:
  umbrel_main_network:
    external: true
```

```
sudo chown root:root \
  /home/umbrel/umbrel/app-data/forgejo-runner/docker-compose.yml
sudo chmod 644 \
  /home/umbrel/umbrel/app-data/forgejo-runner/docker-compose.yml
```

Step 4: Get a registration token from Forgejo

In Forgejo: Site Administration → Actions → Runners → Create new runner. Copy the token.

Step 5: Register the runner

Inline the token directly — do not save it to a script file.

```
cd /home/umbrel/umbrel/app-data/forgejo-runner
sudo docker compose run --rm --user 1001:1001 \
  --entrypoint forgejo-runner runner \
  register \
  --no-interactive \
  --instance http://forgejo_server_1:8101 \
  --token 'YOUR_REGISTRATION_TOKEN_HERE' \
  --name umbrel-runner \
  --labels 'node:docker://node:24-bookworm,\
go:docker://ghcr.io/catthehacker/ubuntu:act-22.04,\
docker:docker://ghcr.io/catthehacker/ubuntu:act-22.04'
```

Expected final line: **Runner registered successfully.** Verify: `sudo cat /home/umbrel/umbrel/app-data/forgejo-runner/data/.runner | head -5`

Step 6: Start the daemon

```
cd /home/umbrel/umbrel/app-data/forgejo-runner
sudo docker compose up -d
sudo docker compose logs --tail=20
```

You want to see: *Starting runner daemon*, *declared successfully*, and *[poller] launched*.

Step 7: Verify in the Forgejo UI

Visit https://forgejo.example.com/-/admin/actions/runners. umbrel-runner should show **Idle** with all three labels.

Step 8: Enable Actions per repository

Repo → Settings → Units (or Advanced Settings) → check Enable Repository Actions → Save.

6.6 Writing Workflows: Action References

Forgejo's DEFAULT_ACTIONS_URL points to data.forgejo.org. Actions that live on GitHub must use the full URL or Forgejo will try to clone them from its own mirror and fail.

Where the action lives	How to reference it
data.forgejo.org/actions/* (checkout, setup-node, setup-go, setup-beam, cache, upload-artifact)	Bare: uses: actions/checkout@v4
GitHub-hosted (superfly, oven-sh, docker, etc.)	Full URL: uses: https://github.com/oven-sh/setup-bun@v2

Workflow examples:

Node / SvelteKit

```
jobs:
  build:
    runs-on: node
    steps:
      - uses: actions/checkout@v4
      - uses: actions/setup-node@v4
        with: { node-version: 24, cache: npm }
      - run: npm ci && npm run build
```

Bun

```
jobs:
  build:
    runs-on: node
    steps:
      - uses: actions/checkout@v4
      - uses: https://github.com/oven-sh/setup-bun@v2
      - run: bun install --frozen-lockfile && bun run build
```

Go

```
jobs:
  build:
    runs-on: go
    steps:
      - uses: actions/checkout@v4
      - uses: actions/setup-go@v5
        with: { go-version: '1.23', cache: true }
      - run: go test ./...
```

Fly.io Deploy

```
jobs:
  deploy:
    runs-on: node
    steps:
      - uses: actions/checkout@v4
      - uses: https://github.com/superfly/flyctl-actions/setup-flyctl@v1.5
```

```
      - run: flyctl deploy --remote-only
        env:
          FLY_API_TOKEN: ${{ secrets.FLY_API_TOKEN }}
```

TIP Pin third-party actions to a release tag (@v1.5) rather than @master. A breaking change in a moving branch should not silently land in your deploys.

6.7 Runner Operations

```
cd /home/umbrel/umbrel/app-data/forgejo-runner
sudo docker compose ps             # check status
sudo docker compose up -d          # start
sudo docker compose restart        # restart
sudo docker compose down           # stop
sudo docker compose logs -f        # tail logs
sudo docker compose pull && \
  sudo docker compose up -d        # update to latest patch
```

6.8 Post-OS-Update Recovery

UmbrelOS updates restart Docker and may stop or remove the runner container. The persistent files in app-data/ survive. Recovery is usually one command:

```
cd /home/umbrel/umbrel/app-data/forgejo-runner
sudo docker compose up -d
sudo docker compose logs --tail=20
# Watch for 'declared successfully' and '[poller] launched'
```

Common post-update issues:

- **docker.sock GID changed** — symptom: permission denied while trying to connect to the Docker daemon socket. Fix: update group_add in docker-compose.yml to match stat -c '%g' /var/run/docker.sock, then restart.
- **umbrel_main_network recreated** — symptom: dial tcp: lookup forgejo_server_1 ... no such host. Fix: docker compose down && docker compose up -d to force re-attachment.
- **Forgejo major version bump** — check runner compatibility before bumping the runner image tag.

You only need a new registration token if data/.runner is deleted or corrupted, or if the runner was manually deleted from the Forgejo admin UI. In all other cases docker compose up -d is sufficient.

6.9 Directory Layout

```
/home/umbrel/umbrel/app-data/forgejo-runner/
■■■ docker-compose.yml          # Runner service definition
■■■ data/                       # UID 1001:1001, mode 775
    ■■■ .runner                 # Registration credentials — SENSITIVE
    ■■■ runner-config.yml       # Runner daemon config
    ■■■ .cache/                 # Cache server data (UID 1001:1001, g+s)
```

7. Migrating Repos from GitHub

Forgejo has a built-in migration tool that can pull repositories from GitHub including commit history, issues, pull requests, labels, milestones, and releases. No manual git remote juggling required.

7.1 Generate a GitHub Personal Access Token

Step 1

- In GitHub, go to Settings → Developer Settings → Personal Access Tokens → Tokens (classic).

Step 2

- Generate a new token with the repo scope (read access is sufficient for migration).

Step 3

- Copy the token — you'll use it once and can revoke it afterward.

7.2 Migrate a Repository

Step 1

- In Forgejo, click the + icon → New Migration.

Step 2

- Select GitHub as the source.

Step 3

- Enter your GitHub username and the Personal Access Token from Step 7.1.

Step 4

- Enter the repository URL (https://github.com/username/repo).

Step 5

- Choose what to migrate: Issues, Pull Requests, Labels, Milestones, Releases.

Step 6

- Click Migrate Repository. For large repos this may take a few minutes.

> *TIP Forgejo can also mirror a GitHub repo, keeping it in sync with the GitHub source. Useful if you want a transition period before fully cutting over.*

7.3 Update Your Local Git Remotes

```
# Check current remote
git remote -v

# Update to Forgejo
git remote set-url origin git@git.example.com:your-username/repo.git

# Verify
git remote -v
```

8. SSH Through Cloudflare Tunnel for Git

Cloudflare Tunnel does not transparently proxy raw TCP. For SSH, the client-side cloudflared binary wraps the session in a WebSocket, sends it through Cloudflare's edge, and the server-side cloudflared unwraps it. Every machine that uses git push/pull/clone over SSH must have cloudflared installed.

8.1 Install cloudflared on Development Machine

```
# macOS
brew install cloudflared

# Debian/Ubuntu
curl -L https://github.com/cloudflare/cloudflared/releases/latest/download/\
cloudflared-linux-amd64.deb -o cloudflared.deb
sudo dpkg -i cloudflared.deb

# Arch
yay -S cloudflared
```

8.2 Configure ~/.ssh/config

```
Host git.example.com
    User git
    ProxyCommand cloudflared access ssh --hostname %h
    IdentityFile ~/.ssh/id_ed25519
    IdentitiesOnly yes
```

8.3 Test the Connection

```
ssh -T git@git.example.com
# Expected: "Hi your-username! You've successfully authenticated..."

git clone git@git.example.com:your-username/my-repo.git
```

On first connection, cloudflared opens a browser window for Cloudflare Access authentication. A token is cached locally for the session duration.

8.4 HTTPS Alternative (No cloudflared Required)

HTTPS clone works natively through the tunnel with zero client-side setup:

```
git clone https://forgejo.example.com/your-username/my-repo.git
```

Authenticate with Forgejo username/password or a personal access token (Forgejo: Settings → Applications). Both SSH and HTTPS can be used simultaneously.

9. Troubleshooting

Error 1033: Cloudflare Tunnel Error

Cloudflare sees the DNS record pointing to your tunnel, but the connector is not connected. The DNS is fine — the problem is on the Umbrel side.

```
# Is the cloudflared container running?
sudo docker ps --format '{{.Names}}\t{{.Status}}' | grep cloud

# Check connector logs
sudo docker logs cloudflared_connector_1 --tail 50

# Verify network membership
sudo docker inspect cloudflared_connector_1 \
  --format '{{range $k,$v := .NetworkSettings.Networks}}{{$k}} {{end}}'

# Test connectivity to Forgejo from cloudflared
sudo docker exec cloudflared_connector_1 \
  wget -qO- http://forgejo_server_1:8101 | head -5
```

Common causes:

- cloudflared container not running — restart via Umbrel dashboard
- Token expired or rotated — re-paste current token in Umbrel app
- Container not on umbrel_main_network — restart the app

docker compose Fails with Blank Variable Warnings

```
WARN[0000] The "APP_DATA_DIR" variable is not set. Defaulting to a blank string.
```

Umbrel injects these variables. Don't run docker compose directly. Use the Umbrel dashboard, or source variables first:

```
export APP_DATA_DIR="/home/umbrel/umbrel/app-data/forgejo"
source /home/umbrel/umbrel/app-data/forgejo/exports.sh
sudo -E docker compose down
sudo -E docker compose up -d
```

Useful Diagnostic Commands

```
# All Forgejo containers and their status
sudo docker ps --format '{{.Names}}\t{{.Status}}\t{{.Ports}}' | grep forgejo

# Forgejo application logs
sudo docker logs forgejo_server_1 --tail 100

# Verify container names
sudo docker ps --format '{{.Names}}' | grep -E 'forgejo|cloud'
```

```
# Check what network containers are on
sudo docker inspect forgejo_server_1 \
  --format '{{range $k,$v := .NetworkSettings.Networks}}{{$k}} {{end}}'

# Verify app.ini is being read correctly
sudo docker exec forgejo_server_1 cat /var/lib/gitea/custom/conf/app.ini
```

10. Remaining Setup Tasks

Cloudflare Dashboard Hardening

- ■ SSL/TLS: set encryption mode to Full (Strict)
- ■ Security → Bots: enable Bot Fight Mode
- ■ Security → WAF: deploy Cloudflare Managed Ruleset
- ■ Security → Settings: enable Block AI Bots
- ■ DNS: enable DNSSEC

Router Verification

- ■ Confirm no port forwarding rules on your home router point to 192.168.x.x (Cloudflare Tunnel is outbound-only — zero inbound ports should be open)

Forgejo Account Security

- ■ Enable two-factor authentication on admin account (Settings → Security)
- ■ Generate and persist a SECRET_KEY if still empty:

```
sudo docker exec forgejo_server_1 forgejo generate secret SECRET_KEY
```

Paste output into the `SECRET_KEY =` line in app.ini, then restart.

Service Token for Git HTTPS Operations

When Cloudflare Access protects forgejo.example.com, bare `git clone https://...` gets redirected to the login page. Service tokens fix this.

- ■ Create a service token: Access → Service Auth → Service Tokens → Create Service Token
- ■ Name: "Git Client", duration: non-expiring (or your preference)
- ■ IMMEDIATELY save Client ID and Client Secret — secret shown only once
- ■ Add a Service Auth policy to the Forgejo-Web application (must be higher priority than Allow Only Me)

Configure Git client:

```
git config --global http.https://forgejo.example.com/.extraHeader \
  "CF-Access-Client-Id: <your-client-id>"
git config --global http.https://forgejo.example.com/.extraHeader \
  "CF-Access-Client-Secret: <your-client-secret>"

# Both headers are additive (not overwritten). Verify with:
git config --global --get-all http.https://forgejo.example.com/.extraHeader
```

For SSH with service token (skips browser popup):

```
Host git.example.com
```

```
User git
ProxyCommand cloudflared access ssh --hostname %h \
  --id <client-id> --secret <client-secret>
IdentityFile ~/.ssh/id_ed25519
IdentitiesOnly yes
```

Development Machine Setup

- ☐ Install cloudflared (see section 8.1)
- ☐ Configure ~/.ssh/config (see section 8.2)
- ☐ Add SSH public key to Forgejo (Settings → SSH / GPG Keys)

Testing Checklist

- ☐ Browser: https://forgejo.example.com (should prompt Cloudflare Access login)
- ☐ ssh -T git@git.example.com (should authenticate via cloudflared)
- ☐ git clone via HTTPS with service token
- ☐ git clone via SSH through cloudflared proxy
- ☐ git push to confirm write access

11. Maintenance Notes

After Umbrel Forgejo App Updates

The docker-compose.yml gets replaced. Re-apply these four env var changes:

```
FORGEJO__server__DOMAIN: forgejo.example.com     # was ${APP_DOMAIN}
FORGEJO__server__SSH_DOMAIN: git.example.com     # was ${APP_DOMAIN}
FORGEJO__server__ROOT_URL: https://forgejo.example.com/  # was not present
FORGEJO__server__SSH_PORT: 22                    # was ${APP_FORGEJO_SSH_PORT}
```

app.ini changes survive updates (persistent data volume).

Security Architecture Summary

The setup provides defense in depth:

- **Network layer** — No port forwarding on the router. Cloudflare Tunnel uses outbound-only connections. Home IP is never exposed.
- **Edge layer** — Cloudflare WAF, Bot Fight Mode, DNSSEC protect against automated attacks.
- **Authentication layer** — Cloudflare Access requires email OTP + geolocation before any traffic reaches the Umbrel.
- **Application layer** — Forgejo requires sign-in to view anything, registration is disabled, 2FA is enabled, session cookies are HTTPS-only.

All traffic flows: Client → Cloudflare Edge (Access check) → Tunnel → Umbrel Docker network → Container. No direct path exists from the internet to the Umbrel.

Resources

- Forgejo Documentation — https://forgejo.org/docs/latest/
- Forgejo Runner Docs — https://forgejo.org/docs/latest/admin/actions/
- Cloudflare Tunnels Docs — https://developers.cloudflare.com/cloudflare-one/connections/connect-networks/
- Cloudflare Zero Trust — https://one.dash.cloudflare.com
- Umbrel — https://umbrel.com
- Blog post — https://codybrunner.com/articles/2026/from-github-to-forgejo/

codybrunner.com · May 2026