

Android Cast build ecosystem

From BUILD_DEPLOY.md

Field	Value
Author	Anton Afanasyeu
Revision	R1
Creation date	2026-05-30
Last modification date	2026-05-30
Co-authored	
Severity	medium
State	in progress
Document type	technical

Table of contents

- [Shared authentication](#)
- [Local stack](#)
- [Local developer rebuild](#)
- [Trigger a build](#)
- [Docker CI](#)
- [CircleCI](#)
- [Production notes](#)
 - [Alpine BE \(config.php paths\)](#)

Source: `BUILD_DEPLOY.md` — section links work in PDF viewers that support internal anchors.

Shared authentication

All web consoles (Crashes, Tickets, Builder) share:

Setting	Value
Session name	ac_crash_sess
Cookie path	/app/androidcast_project
User store	MariaDB users table (same DB as crashes)
PHP auth	examples/crash_reporter/backend/src/Auth.php

Sign in once on any console; the session cookie is valid for /crashes/, /build/, and future apps under the same path prefix.

Orchestration generates matching configs:

- orchestration/runtime/crash-config.php
- orchestration/runtime/build-config.php

Both use `${MARIADB_USER}` / `${MARIADB_PASSWORD}` from `orchestration/.env`.

Local stack

```
cd orchestration
./deploy.sh
```

URL	Purpose
http://localhost:8080/app/androidcast_project/	Hub landing
.../crashes/	Crash / ticket console
.../build/	APK builder console
.../v0/ota/	Published OTA artifacts (when auto-deploy enabled)

Direct debug ports: crashes :8082, builder :8083.

Local developer rebuild

`./rebuild.sh` (repo root):

1. `scripts/init-third-party-submodules.sh` — libvpx, opus, speex, **wireguard-android** (+ nested `wireguard-tools` / `elf-cleaner`)
2. `scripts/build-native-codecs.sh` per ABI
3. `./gradlew :tunnel:assembleDebug :app:assembleDebug` — WireGuard userspace (`libwg-go.so`) from `third-party/wireguard-android`
4. Unit tests, `adb install` (when devices configured)

No runtime fetch from `zx2c4/GitHub` during Gradle: sources come from checked-out `third-party/` submodules only.

Trigger a build

1. Open Builder → sign in (default dev user from crash schema seed).
2. Set git ref, pipeline options, OTA channel.
3. **Start build** → detail page polls log every 1s.
4. Artifacts land in `out/builds/<id>/`; optional OTA copy to `orchestration/runtime/ota-artifacts/`.

Docker CI

- Image version label: `00.BB.CC.DDDD` (see root `Dockerfile`).
- Host wrapper: `scripts/docker-build-runner.sh`
- In-container pipeline: `scripts/ci-build-and-publish-ota.sh`
- OTA generator: `scripts/generate-ota-v0.sh` [`channel`]

CircleCI

- Pipeline description: `build.config.yml`
- Local CLI stub: `.circleci/config.yml` (`circleci config validate`)

Production notes

- Mount builder PHP with Docker socket (build host only).
- FE nginx: read-only OTA under `/v0/ota/`; BE RWX on artifact mount.
- Keep `session_cookie_path` identical across all PHP apps behind the hub.

Alpine BE (config.php paths)

Key	Example on artc0
-----	------------------

build.repo_root	/var/www/.../androidcast_project/android_cast (git tree)
build.artifacts_root	/var/www/.../androidcast_project/out/builds — must exist , writable by PHP-FPM (nginx)
build.ota_mount	/var/www/.../androidcast_project/ota-artifacts — nginx serves .../v0/ota/ at /v0/ota/
build.runner_script	full path to scripts/docker-build-runner.sh (no ... placeholders)

```

sudo examples/crash_reporter/backend/scripts/prepare-be-ota-mount.sh
mkdir -p .../out/builds
chown -R nginx:nginx .../out .../ota-artifacts
chmod -R 775 .../out .../ota-artifacts

```

Apply nginx OTA block: nginx.apps-port80.fragment (location ^~ /v0/ota/), then nginx -t && rc-service nginx reload on BE.

If **Start build** shows *Network error*, check BE php-fpm81 log: often mkdir(): Permission denied on out/builds (HTTP 500, not a browser network fault).