LARAVEL CHEAT SHEET

Deployer for Laravel

Build once in CI, smoke-test the candidate, roll back once on its own. Deployer 8, one VM, Nginx + PHP-FPM.

TL;DR

- Keep Deployer for release dirs, shared files, the lock, the atomic switch and cleanup.
- Ship a **CI-built tarball** checked by SHA-256. No Git, Composer or Node on the server.
- Migrations run only with `DEPLOY_ALLOW_MIGRATIONS=true`, and must be additive.
- Smoke-test the new release **before** `deploy:symlink`, readiness **after** it.
- Failed readiness after the switch: **one** automatic rollback, then the deploy fails.

What the stock recipe runs

`recipe/laravel.php`, Deployer 8 (v8.0.5 as of 2026-10-27).

```
deploy:prepare        # info, setup, lock, release, update_code,
                      # env, shared, writable
deploy:vendors        # composer install on the server
artisan:storage:link
artisan:optimize
artisan:migrate       # every deploy, unconditionally
deploy:publish        # symlink, unlock, cleanup, success
artisan:reload
```

Redefining a task with the same name replaces its body. Keep the groups, swap `deploy:update_code` and `deploy:vendors`, write your own `deploy` list.

Layout and settings

```
/var/www/starter/
├── .dep/                  # lock, release log, state files
├── current -> releases/42
├── releases/  38/ … 42/
└── shared/    .env  storage/
```

```
set('keep_releases', 5);
set('shared_files', ['.env']);
set('shared_dirs', ['storage']);
set('writable_dirs', ['bootstrap/cache', 'storage']);
set('writable_mode', 'acl');
```

Host, user, path, artifact and its SHA-256 come from env vars (`DEPLOY_HOST` , `DEPLOY_USER` , `DEPLOY_PATH` , `DEPLOY_ARTIFACT` , `DEPLOY_ARTIFACT_SHA256`). Missing one = the recipe refuses to start.

Upload the artifact

```
task('deploy:update_code', function (): void {
    $artifactPath = get('artifact_path');

    if (! is_file($artifactPath)) {
        throw new \RuntimeException(sprintf('Release artifact not found: %s.', $artifactPath));
    }

    upload($artifactPath, '{{release_path}}/release.tar.gz');
    run('test "$(sha256sum {{release_path}}/release.tar.gz | cut -d " " -f 1)" = "
{{artifact_sha256}}"');
    run('tar -xzf {{release_path}}/release.tar.gz -C {{release_path}} --no-same-owner');
    run('rm {{release_path}}/release.tar.gz');
    run('printf "%s\n" "{{release_name}}" > {{release_path}}/RELEASE');
});
```

CI packs `git archive` + `vendor/` (`--no-dev --classmap-authoritative`) + `public/build` + `bootstrap/ssr` . The checksum is compared on the server, after upload. `deploy:vendors` only runs `test -f {{release_path}}/vendor/autoload.php` .

Before the switch

```
task('deploy:migrate', function (): void {
    if (getenv('DEPLOY_ALLOW_MIGRATIONS') !== 'true') {
        writeln('<comment>Migrations skipped: DEPLOY_ALLOW_MIGRATIONS=true is required.</comment>');

        return;
    }

    run('cd {{release_path}} && php artisan migrate --force --no-interaction');
});
```

Expand/contract only. The schema is never rolled back: a bad migration gets a forward fix.

```
run('test -f {{release_path}}/public/build/manifest.json');
run('test -f {{release_path}}/bootstrap/ssr/app.js');
run('cd {{release_path}} && php artisan about --only=environment --no-interaction');
run('cd {{release_path}} && php artisan ops:readiness --no-interaction');
```

`deploy:smoke` : boots the framework with the release's cached config. `ops:readiness` is the starter's own command (DB, Redis, storage).

Switch and reload

```
root /var/www/starter/current/public;

location = /index.php {
    include fastcgi_params;
    fastcgi_pass unix:/run/php/starter.sock;
    fastcgi_param SCRIPT_FILENAME $realpath_root/index.php;
    fastcgi_param DOCUMENT_ROOT $realpath_root;
}
```

`deploy:symlink` = `mv -T release current` (atomic). `$realpath_root` keeps OPcache honest without reloading PHP-FPM.

```
run('cd {{current_path}} && php artisan horizon:terminate --no-interaction');
run('cd {{current_path}} && php artisan schedule:interrupt --no-interaction');
run('cd {{current_path}} && (php artisan inertia:stop-ssr --no-interaction || true)');
```

systemd units with `WorkingDirectory=…/current` and `Restart=always` come back on the new release. Queued jobs must work across one release.

One rollback, never two

- `deploy:remember_healthy`: `readlink -f current` into `.dep/healthy_release` before the switch.
- `deploy:smoke:live`: `/health/ready` via `curl --resolve host:443:127.0.0.1`, 3 attempts, 5 s timeout, 5 s apart.
- Fails: switch back, write `BAD_RELEASE`, restart workers, check again.
- Old release fails too: write `.dep/deploy_blocked`. Every later deploy stops until a person deletes it.
- Every branch throws, so the workflow run is red.

```
run('cd {{deploy_path}} && ln -sfn '.escapeshellarg($healthy).' current.rollback && mv -T
current.rollback current');
```

Atomic, like the forward switch. The built-in `rollback` task uses `ln -nfs` directly, and checks `BAD_RELEASE` before picking a candidate.

The task list

```
'deploy:check_blocked',
'deploy:prepare',
'deploy:vendors',
'artisan:storage:link',
'artisan:config:cache',
'artisan:route:cache',
'artisan:view:cache',
'artisan:event:cache',
'deploy:migrate',
'deploy:smoke',
'deploy:remember_healthy',
```

Above the switch: can fail without users noticing.

```
'deploy:symlink',
'deploy:restart_workers',
'deploy:smoke:live',
'deploy:unlock',
'deploy:cleanup',
'deploy:success',

after('deploy:failed', 'deploy:unlock');
```

Below the switch: covered by the rollback. Check the order with `dep tree deploy`.

From GitHub Actions

```
export DEPLOY_ARTIFACT_SHA256="$(cut -d ' ' -f 1 .release/starter.tar.gz.sha256)"
vendor/bin/dep deploy "$DEPLOY_ENVIRONMENT" --file=deploy.php --revision="$GITHUB_SHA" --ansi
```

- Verify the artifact's SHA-256 and that the manifest commit equals `GITHUB_SHA`.
- `workflow_dispatch` only; `concurrency: deploy-${{ inputs.environment }}`, `cancel-in-progress: false`.
- SSH key as an environment secret, pinned `known_hosts`.
- Required reviewers on `production` (check your GitHub plan for private repos).
- `deployer/deployer: ^8.0` in `require-dev`.

When not to use this

- **Containers:** the image is the artifact, the orchestrator switches and rolls back.
- **Several web servers behind a load balancer:** the switch is no longer atomic across hosts.
- **A managed platform** (Laravel Cloud, Forge with Envoyer) already does it.
- **Non-additive migrations:** no tool gives zero downtime. Plan a maintenance window.

RECAP

The sheet in short

- Build in CI, upload a tarball, check SHA-256 on the server
- Migrations only on an explicit flag, expand/contract only
- Smoke the candidate before `deploy:symlink`
- `$realpath_root` instead of reloading PHP-FPM
- Restart Horizon, scheduler and SSR after the switch
- One automatic rollback, then block and fail loudly

Found this useful?

Save this cheat sheet for later.

Full article: dominik-dev.pl

Share • Comment