

No Cache Detected

Why This Error Occurred

A Next.js build was triggered in a continuous integration environment, but no cache was detected.

This results in slower builds and can hurt Next.js' persistent caching of client-side bundles across builds.

Possible Ways to Fix It

Note: If this is a new project, or being built for the first time in your CI, you can ignore this error. However, you'll want to make sure it doesn't continue to happen and fix it if it does!

Configure Next.js' cache to be persisted across builds. Next.js stores its cache in the `.next/cache` directory.

Storing this folder across builds varies by CI provider. We've provided a list of a few common providers below.

Vercel Next.js caching is automatically configured for you. There's no action required on your part.

CircleCI Edit your `save_cache` step in `.circleci/config.yml` to include `.next/cache`:

```
steps:
  - save_cache:
      key: dependency-cache-{{ checksum "yarn.lock" }}
      paths:
        - ./node_modules
        - ./next/cache
```

If you do not have a `save_cache` key, please follow CircleCI's documentation on setting up build caching.

Travis CI Add or merge the following into your `.travis.yml`:

```
cache:
  directories:
    - $HOME/.cache/yarn
    - node_modules
    - next/cache
```

GitLab CI Add or merge the following into your `.gitlab-ci.yml`:

```
cache:
  key: ${CI_COMMIT_REF_SLUG}
  paths:
    - node_modules/
    - .next/cache/
```

Netlify CI Use Netlify Plugins with `@netlify/plugin-nextjs`.

AWS CodeBuild Add (or merge in) the following to your `buildspec.yml`:

```
cache:
  paths:
    - 'node_modules/**/*' # Cache `node_modules` for faster `yarn` or `npm i`
    - '.next/cache/**/*' # Cache Next.js for faster application rebuilds
```

GitHub Actions Using GitHub's `actions/cache`, add the following step in your workflow file:

```
uses: actions/cache@v2
with:
  # See here for caching with `yarn` https://github.com/actions/cache/blob/main/examples.md
  path: |
    ~/.npm
    ${ github.workspace }/.next/cache
  # Generate a new cache whenever packages or source files change.
  key: ${ runner.os }-nextjs-${ hashFiles('**/package-lock.json') }-${ hashFiles('**.[
  # If source files changed but packages didn't, rebuild from a prior cache.
  restore-keys: |
    ${ runner.os }-nextjs-${ hashFiles('**/package-lock.json') }-
```

Bitbucket Pipelines Add or merge the following into your `bitbucket-pipelines.yml` at the top level (same level as `pipelines`):

```
definitions:
  caches:
    nextcache: .next/cache
```

Then reference it in the `caches` section of your pipeline's `step`:

```
- step:
  name: your_step_name
  caches:
    - node
    - nextcache
```

Heroku Using Heroku's custom cache, add a `cacheDirectories` array in your top-level `package.json`:

```
"cacheDirectories": [".next/cache"]
```

Azure Pipelines Using Azure Pipelines' Cache task, add the following task to your pipeline yaml file somewhere prior to the task that executes `next build`:

```
- task: Cache@2
  displayName: 'Cache .next/cache'
  inputs:
    key: next | $(Agent.OS) | yarn.lock
    path: '$(System.DefaultWorkingDirectory)/.next/cache'
```