

[← Back to Blog](#)

Wednesday, April 9th 2025

Next.js 15.3

Posted by

**Andrew Clark**
@acd lite**Jiwon Choi**
@devjiwonchoi**Jude Gao**
@gao_jude**Maia Teegarden**
@padmaia**Tim Neutkens**
@timneutkens**Will Binns-Smith**
@wbinssmith

Next.js 15.3 includes Turbopack for builds, new client instrumentation and navigation hooks, and more:

- **Turbopack for builds (alpha)**: Faster production builds passing 8000+ tests (99%)
- **Community support for Rspack (experimental)**: Alternative bundler with Webpack compatibility
- **Client Instrumentation hook**: Early monitoring and analytics setup
- **Navigation hooks**: Control routing with `onNavigate` and `useLinkStatus`
- **TypeScript plugin improvements**: Improved support for large codebases

Upgrade today, or get started with:

>_ Terminal



```
1 # Use the automated upgrade CLI
2 npx @next/codemod@canary upgrade latest
3
4 # ...or upgrade manually
5 npm install next@latest react@latest react-dom@latest
```

```
6
7 # ...or start a new project
8 npx create-next-app@latest
```

Turbopack Builds (alpha)

Following the stable release of `next dev --turbo`, **over 50% of development sessions** on Next.js 15 are now using Turbopack.

This release includes our **alpha** release of `next build --turbo`, bringing the same performance improvements from local development to production builds.

Try Turbopack for production builds by upgrading to 15.3 and running:

>_ Terminal



```
next build --turbo
```

Functionality

99.3% [↗](#) of integration tests for `next build` are already passing. You can track our progress towards 100% at areweturboboyet.com [↗](#). If your application already works with Turbopack for `dev`, then it should work as-is with `build`.

Turbopack builds are in alpha. We don't recommend using them in production for mission-critical applications at this stage. Instead, try them in a preview or staging environment, or run the build locally to observe differences in bundle size and performance.

We're actively working to close these performance gaps through scope hoisting, improved chunking, and other optimizations.

Build performance

We've been validating Turbopack on Vercel's large internal monorepo and early partner codebases. One advantage of Turbopack's architecture versus our previous Webpack

implementation is that performance scales when adding CPU cores:

- At 4 cores: 28% faster than Webpack
- At 16 cores: 60% faster than Webpack
- At 30 cores: 83% faster than Webpack

These build times drop even further with our experimental work on persistent caching. We will share more on that in a future release.

Ecosystem

We're working with commonly used integrations like Sentry to make sure they're compatible with `next build --turbopack` before the stable release. Please reach out to [@leerob on X](#) if you are a tool author who would like to work with us to ensure compatibility.

Feedback

Please share your feedback, even if it goes smoothly, to help us prepare a stable release:

- [GitHub discussion](#) s for general feedback
- [GitHub issues](#) for reproductions

Turbopack configuration in `next.config.ts` (stable)

Turbopack configuration in `next.config.ts` has moved from `experimental.turbo` to the top-level `turbopack` key:

`TS` next.config.ts



```
1 import type { NextConfig } from 'next';
2
3 const nextConfig: NextConfig = {
4   turbopack: {
5     rules: {
6       '*.svg': {
```

```
7      loaders: ['@svgr/webpack'],
8      as: '*.js',
9    },
10  },
11  },
12  };
13
14  export default nextConfig;
```

For compatibility, the `experimental.turbo` option will continue to be supported until the next major release of Next.js.

For a complete list of Turbopack configuration options, see the [Turbopack API Reference](#).

Community support for Rspack (experimental)

The Rspack team has created a community plugin for Next.js.

This provides an option for Next.js users who need near-exact Webpack API compatibility, but cannot yet move to Turbopack, to improve their local compilation and build times. We're confident in our progress with Turbopack and will continue to provide an incremental path forward for Webpack users.

While this is not an official Next.js plugin, we are partnering with the Rspack team. Both teams will collaborate on shared foundations like SWC and Lightning CSS, benefiting all Next.js users and the broader ecosystem.

If you want to explore using Rspack with Next.js, you can use the `next-rspack` adapter. We are running the adapter against our integration test suite. It currently passes ~96% of tests.

[View an example](#) or learn more in the [Rspack docs](#).

Client Instrumentation Hook

The `instrumentation-client.js|ts` file allows you to add monitoring and analytics code that runs before your application's frontend code starts executing.

This is ideal for setting up performance tracking, error monitoring, or other client-side observability tools as early as possible in the lifecycle.

`instrumentation-client.js`

```
1 // Set up performance monitoring
2 performance.mark('app-init');
3
4 // Initialize analytics
5 console.log('Analytics initialized');
6
7 // Set up error tracking
8 window.addEventListener('error', (event) => {
9   // Send to your error tracking service
10   reportError(event.error);
11 });
```

Place this file at the root of your project similar to [server-side instrumentation](#).

Learn more in the [instrumentation-client file documentation](#).

Navigation Hooks

We're introducing new navigation hooks that enhance client-side routing capabilities in Next.js 15.3, allowing you to more easily develop localized loading states and implement complex controls like navigation cancelation.

`onNavigate`

This event handler is a new property of the `Link` component and executes during client-side navigations, giving you precise control over your application's routing behavior.

Unlike the `onClick` event, which fires for all clicks, `onNavigate` can be used for [Single-Page App \(SPA\)](#) navigations, allowing you to execute code or even cancel navigation with `preventDefault()`.

This API can be used to implement transition animations, navigation guards, or analytics tracking that should only run during actual page transitions.

Learn more by visiting the [onNavigate documentation ↗](#).

`useLinkStatus`

The `useLinkStatus` Client Component hook returns a `pending` boolean that indicates when navigation is in progress, giving you localized control over loading states.

This API is modeled after [useFormStatus ↗](#) from React, and is helpful for adding custom loading indicators during page transitions, especially when prefetching is disabled or when your linked routes don't have dedicated loading states.

By placing a component that uses `useLinkStatus` as a descendant of your `<Link>` component, you can create responsive UI elements that react to navigation events in real-time.

Learn more by visiting the [useLinkStatus documentation ↗](#).

TypeScript Plugin Performance Improvements

The Next.js TypeScript language server plugin (LSP) is now faster.

The LSP provides inline Intellisense features such as server/client boundary validation, component prop hints, configuration autocompletion, and error detection for disallowed APIs in React Server Components. In very large codebases, the plugin could previously cause the TypeScript language service to hang or crash.

We've made significant performance improvements to resolve these issues. In our internal testing, plugin response times have improved ~60% with no freezing or crashes.

Other Changes

- **[Feature]** Support `new URL()` in `images.remotePatterns` ([#77692 ↗](#))
 - **[Feature]** Viewport options are now separate from `metadata` ([#77427 ↗](#))
 - **[Feature]** Add `unstable_dynamicOnHover` option ([#77866 ↗](#))
 - **[Feature]** Add support for Pinterest Rich Pins ([#76988 ↗](#))
 - **[Improvement]** Make `revalidate` work when followed by a redirect in Route Handlers ([#77090 ↗](#))
 - **[Improvement]** Ensure strong consistency after calling `revalidate` in Server Actions ([#76885 ↗](#))
 - **[Improvement]** Upgrade `sharp` for faster PNG to AVIF conversion ([#77839 ↗](#))
-

Contributors

Next.js is the result of the combined work of over 3,000 individual developers. This release was brought to you by:

- The **Next.js** team: [Andrew ↗](#), [Hendrik ↗](#), [Janka ↗](#), [Jiachi ↗](#), [Jimmy ↗](#), [Jiwon ↗](#), [JJ ↗](#), [Josh ↗](#), [Jude ↗](#), [Sam ↗](#), [Sebastian ↗](#), [Sebbie ↗](#), [Wyatt ↗](#), and [Zack ↗](#).
- The **Turbopack** team: [Benjamin ↗](#), [Donny ↗](#), [Josh ↗](#), [Maia ↗](#), [Niklas ↗](#), [Tim ↗](#), [Tobias ↗](#), and [Will ↗](#).
- The **Next.js Docs** team: [Delba ↗](#), [Rich ↗](#), [Ismael ↗](#), and [Lee ↗](#).

Huge thanks to [@raunofreiberg](#), [@huozhi](#), [@ijjk](#), [@timneutkens](#), [@gaojude](#), [@leerob](#), [@mezotv](#), [@bgw](#), [@samcx](#), [@ztanner](#), [@sokra](#), [@mischnic](#), [@wbinssmith](#), [@kdy1](#), [@unstubbable](#), [@ahabhgk](#), [@ScriptedAlchemy](#), [@SukkaW](#), [@wyattjoh](#), [@eps1lon](#), [@Amirroid](#), [@Netail](#), [@lubieowoce](#), [@gnoff](#), [@jackwilson323](#), [@acd lite](#), [@sbougerel](#), [@kevva](#), [@kasperpeulen](#), [@Cy-Tek](#), [@dvoytenko](#), [@husseinraoouf](#), [@isBatak](#), [@iamkd](#), [@delbaoliveira](#), [@jantimon](#), [@padmaia](#), [@Bernardoow](#), [@styfle](#), [@devjiwonchoi](#), [@JamBalaya56562](#), and [@Marukome0743](#) for helping!



Resources

[Docs](#)

[Support Policy](#)

[Learn](#)

[Showcase](#)

[Blog](#)

[Team](#)

[Analytics](#)

[Next.js Conf](#)

[Previews](#)

More

[Next.js Commerce](#)

[Contact Sales](#)

[Community](#)

[GitHub](#)

[Releases](#)

[Telemetry](#)

[Governance](#)

About Vercel

[Next.js + Vercel](#)

[Open Source Software](#)

[GitHub](#)

[Bluesky](#)

[X](#)

Legal

[Privacy Policy](#)

Subscribe to our newsletter

Stay updated on new releases and features, guides, and case studies.

[Subscribe](#)

© 2025 Vercel, Inc.

