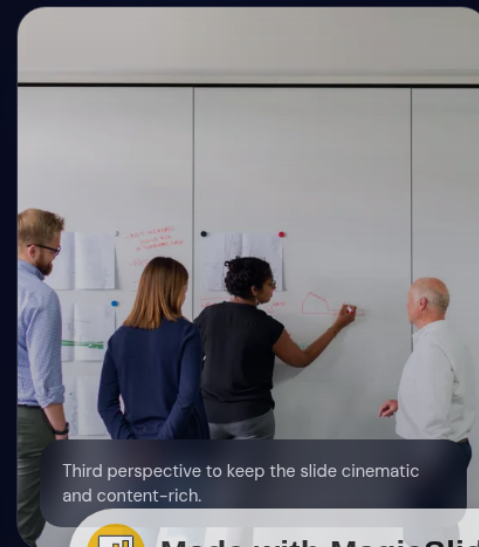
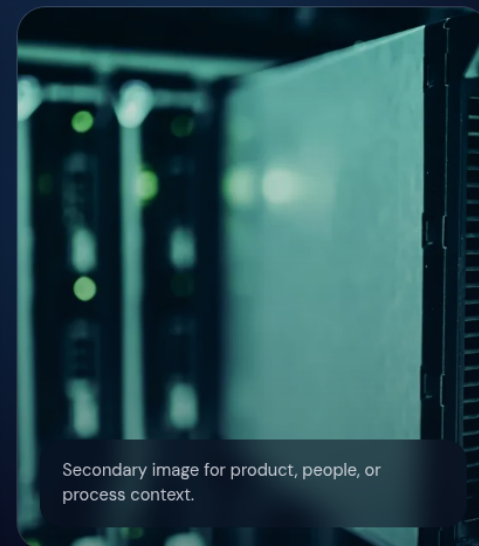


Cutting CI From 22 Minutes to 7

How the Northwind platform team (a fictional company, northwind.example) made every pull request check run three times faster. Same test coverage, smaller bill.

[GitHub Actions](#)[Test sharding](#)[Build cache](#)

● Built for image-led stories, dense insights, and motion-first reveals.

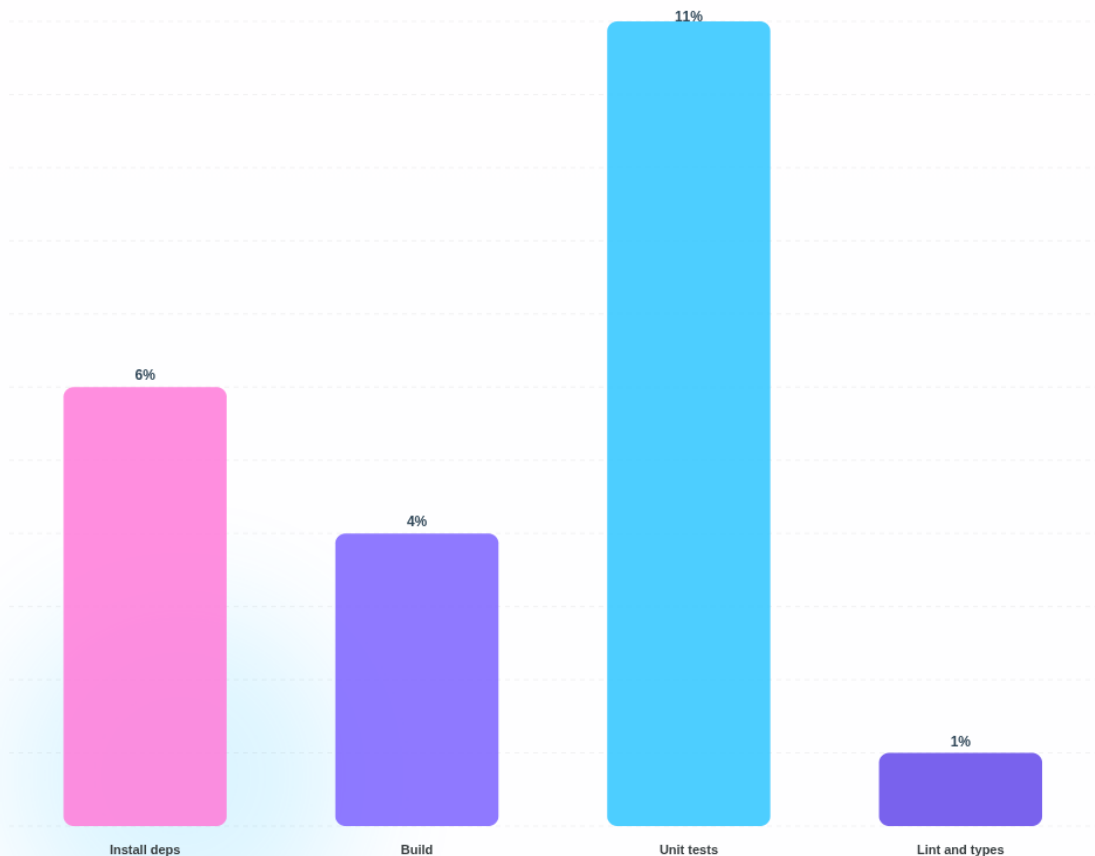


Made with MagicSlides

Median PR pipeline, before and after

Minutes per stage (before)

TOTAL 22 MIN



Minutes per stage, median of 1,400 pull requests in September.

Core Analysis

Tests were half

One runner ran 3,800 tests in series.

No cache

Every run downloaded the same 900 MB of packages.

Builds repeated

Unchanged packages were rebuilt on every push.



WHAT WE CHANGED

Four changes over three sprints

Each change shipped on its own branch and was measured for a week before the next.



SPRINT 1

Cache dependencies

Keyed the package cache on the lockfile hash. Install dropped from 6 to 1 minute.



SPRINT 2

Remote build cache

Only packages touched by the diff rebuild now.



SPRINT 3

Shard tests by 6

Split by past timing data so each shard finishes within 20 seconds of the others.



SPRINT 4

Quarantine flaky tests

41 flaky tests moved to a nightly job with an owner each.

Median pipeline is now **7 minutes**, and reruns fell from 1 in 6 to 1 in 40.

SHIPPED

Made with MagicSlides

RESULTS

Before and after, side by side

Numbers are medians over the 30 days before and after the work.

METRIC	BEFORE	AFTER	CHANGE
PR pipeline	22 min	7 min	-68%
Reruns per PR	1 in 6	1 in 40	-85%
Runner minutes / month	61,000	38,000	-38%
PRs merged / week	212	259	+22%

Sharding uses more runners at once, but **total minutes still fell** because cache hits skip most work.

30-DAY MEDIAN

Made with MagicSlides



LESSONS

What we would do again

Measuring first saved us from tuning the wrong stage. The lint step everyone blamed took one minute.

Measure per stage

Export job timings to a dashboard before changing anything.

One change at a time

A week of data per change made the wins easy to prove.

Give flaky tests owners

Quarantine without an owner just hides the problem.

The fastest test is the one that does not need to run on this diff.

Made with MagicSlides

NEXT QUARTER

Where the next minutes come from

Test impact analysis

Run only tests linked to changed files.

Bigger runners

Trial 8-core runners for the build step.

Target: 5 min

Docker layer cache

Stop rebuilding base images on each push.

Merge queue

Batch main-branch checks during busy hours.

Each item gets **one owner and one week** of measurement, same as before.

Made with MagicSlides