**Week 16**

**16.1 Monorepo**

* apps and packages are common convention in monorepos.
* CI/CD – continuous integration and continuous deployment. Any code change on GitHub gets propagated to AWS.
* Monorepos lets you pull code from different folders without pushing to npm.
* Turborepo is not a monorepo framework but lets you choose one. It’s the best.
* Vite and tsc are good examples of build system.
* Lerna, nx, yarn workspaces are examples of monorepos.
* Build orchestration lets you schedule the builds and parallelize them. First build common, then backend and frontend.
* npx create-turbo@latest
* common == packages
* Whatever you want to export from packages/ui, you need to mention in the exports section in package.json the things that packages/ui retuns.
* You can see that web and docs have @repo/ui as a dependency in their package.json.
* It uses a cache if the files are unchanged.

**16.2 Monorepo**

* After doing npm run build the servers, the HTMLs, and the CSSs are stored in the .next folder.
* Turbo build:
  + I call things parallel.
  + If things are dependent on each other, I first call the child and then the parent.
  + I cache the output of the builds.
* Do npm install in the root folder.
* The package.json for new repositories can be extended from packages/typescript-config
* Adding this "dependencies": {

"@repo/common": "\*"

} in the package.json of backend folder and then doing npm i brings the common folder in the global node\_modules.

* tsc did not work in the, we shifted to esbuild. Another typescript compiler.