New CI Tool: TeamCity

- Boasts “On-the-fly” build progress reporting
- Integrates with all popular build and testing tools
- Config files written in Kotlin (They make it sound like a plus)
- User management with audits, authentication, groups, and roles
- Has a cloud option as well as a download option to run TeamCity on your machine

Getting Started with TeamCity:

- This took me a while to find, so I wasn’t super impressed: https://www.jetbrains.com/help/teamcity/quick-setup-guide.html#TeamCity+First+Start
- There’s a download link and some setup instructions. From what I read, the machine you used to download it can then be used as the TeamCity server for your company/department
- There is a commandline command that turns on the server and a build agent.
- Then when you “open TeamCity in a browser for the first time,” there are some additional steps required to finalize the installation.
- You will then be prompted to create an Admin account.

Configure Your First Project

- Click ‘Administration’, and click “Create project”
- You can import a project from GitHub or make one manually
- You must register TeamCity on GitHub to do this (which has its own steps)
- If you decide to import a project manually, you can follow the steps outlined in the GUI to connect TeamCity to VSCode.
- Either way, you add a build step and choose your build runner (like Docker, Maven, Gradle)

Run Your First Build

- You click “Run”. It did not talk about adding a .yml file or anything like that.
- I assume you should have a project.test.js file with your tests ready to be read by TeamCity

TeamCity is made by JetBrains. It was listed as 3rd in this article (https://www.katalon.com/resources-center/blog/ci-cd-tools/) of top 14 CI/CD tools for 2021. It has been around since at least 2007.
Twitter, eBay, and Wikipedia are reported to use TeamCity (https://www.upguard.com/blog/teamcity-vs-jenkins-for-continuous-integration#:~:text=8.,Tumblr%2C%20and%20Netflix%20among%20them.).
From the same article just above, TeamCity offers professional support services.

New APM/Real-Time Error Monitoring: AppOptics

- Boasts “live code profiling,” which will pinpoint the exact spot in your code that is causing the error.
- Views end-to-end behavior of transactions, finding the bottlenecks that are slowing your program down

- Monitors across many different programming languages, with no manual instrumentation or configuration required.

Getting Started with AppOptics:

- https://documentation.solarwinds.com/en/success_center/appoptics/content/appoptics_getting_started_guide.htm
- Install a Snap Agent
- Install an Application Monitoring Agent
- Read through the AppOpticsAPI to create your own “collector” or use basic scripts
- Create a token and POST metrics
- Voila! You are done.

Beyond Getting Started

- This section was about stuff in their UI.
- I’m guessing all the code that should be written in our project files is located in their API Documentation.
- Even though it might sound annoying at first to have to look up the API info, it doesn’t to me. For some reason, learning about AppOptics was more appealing to me than reading about TeamCity was. I definitely felt that AppOptics’s getting started documentation was easier to find and read.

AppOptics used to be called TraceView (and Tracelytics previously to that), but it was bought by SolarWinds back in 2016.

From this review page (https://www.trustradius.com/products/solarwinds-appoptics/reviews), it seems that AppOptics is being used by smaller companies, but it has an 8.6 out of 10 rating, so people seem to like it.