Jim Lamb

Full-stack Designer

Microsoft

Azure DevOps

Track-It!

Norton by Symantec

I’m a user experience designer, product strategist, and information architect.

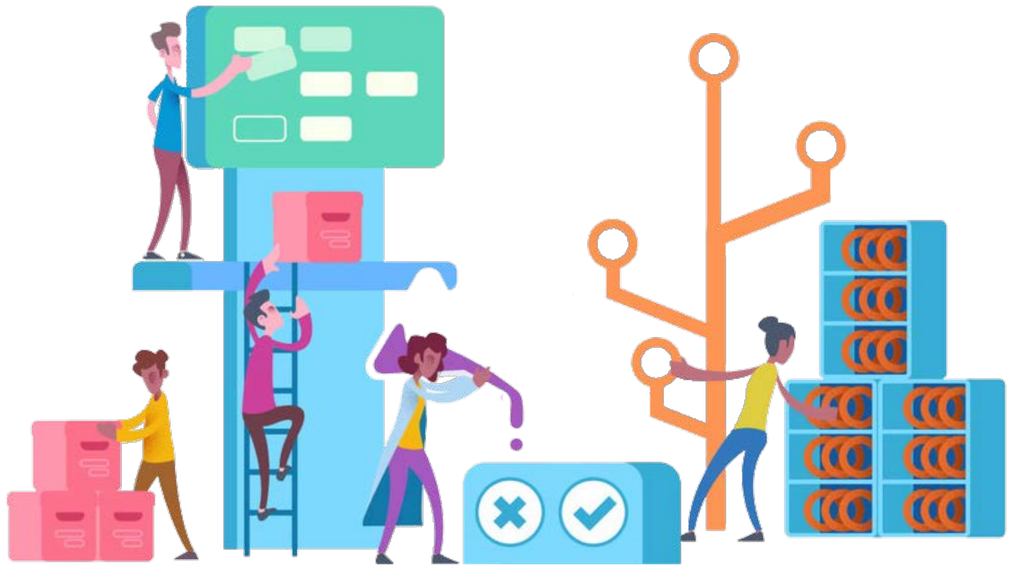
I help teams create products and experiences that solve real user problems.

I use flow mapping, storyboarding, and prototyping to inform and guide the product design process.

About Azure DevOps

Microsoft’s end-to-end DevOps toolchain for planning, developing and deploying software with over a million long-term engaged users.

As the primary designer for Azure Pipelines, I worked with a team of eleven designers and one researcher supporting over 300 technical staff.



The Challenge

- ✔ Simplify adoption of Azure Pipelines for GitHub users.
- ✔ Provide a streamlined authentication experience across the two services.
- ✔ Support configuration-as-code.

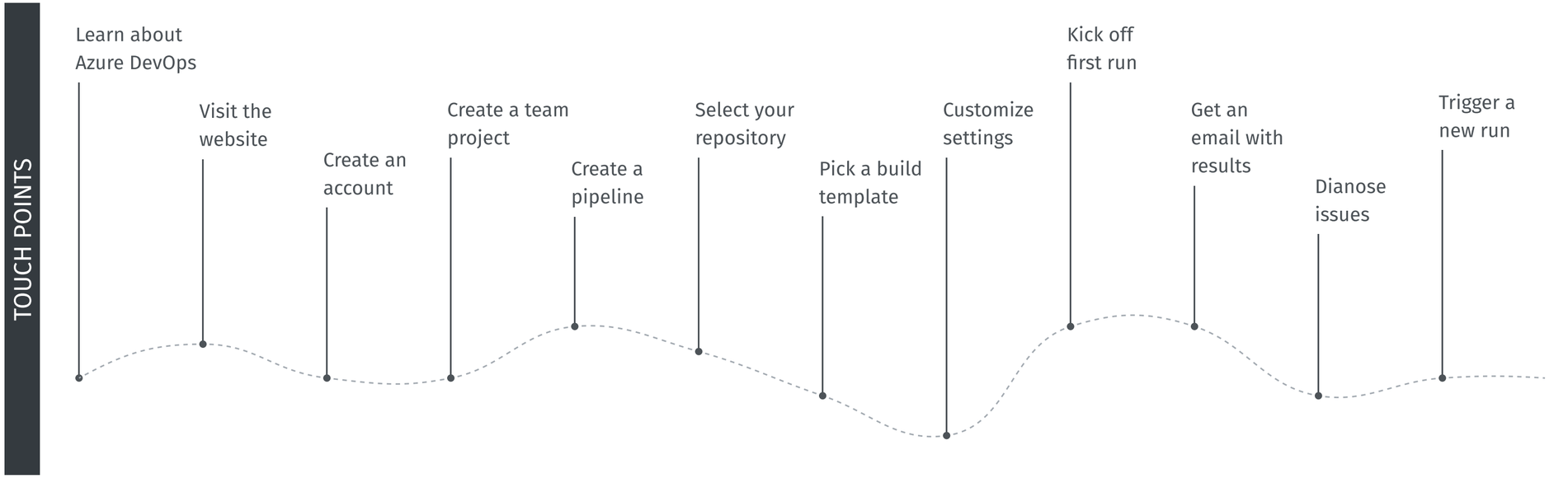
My Role

I was the lead designer for this project, supporting a team of about 12 engineers with one program manager over six months.

I also worked closely with the design system team as we defined new patterns and components to support this experience.

Journey map

I mapped out the existing experience for GitHub users getting started with Azure Pipelines. There were a number of areas for improvement.

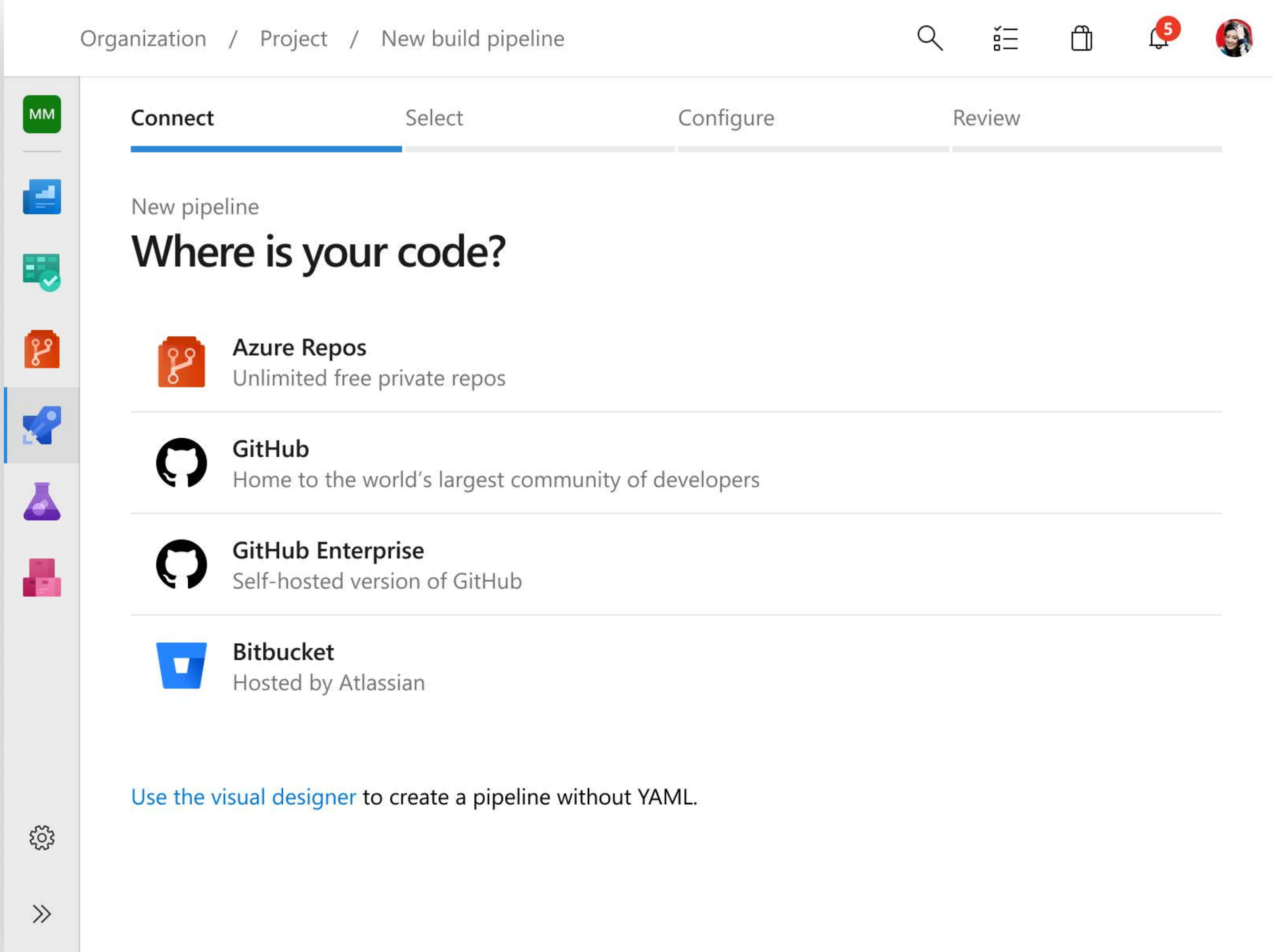


Sign up	Select	Configure	Review
DELIGHTS			
Marketing website looks trustworthy. Microsoft and Azure branding lends credibility.	Marketing website looks trustworthy.	The visual designer is approachable and easy to understand.	Receiving an email notification is helpful, especially for long-running builds.
OPPORTUNITIES			
Create an offering that appears in GitHub's marketplace.	Find a way to bypass these steps - they don't match the user's mental model.	It would be great to already have this context. Setting up authentication is challenging.	Can we do anything to help the user to pick the right template?	Diagnosing failures can be challenging.

Connect to host service

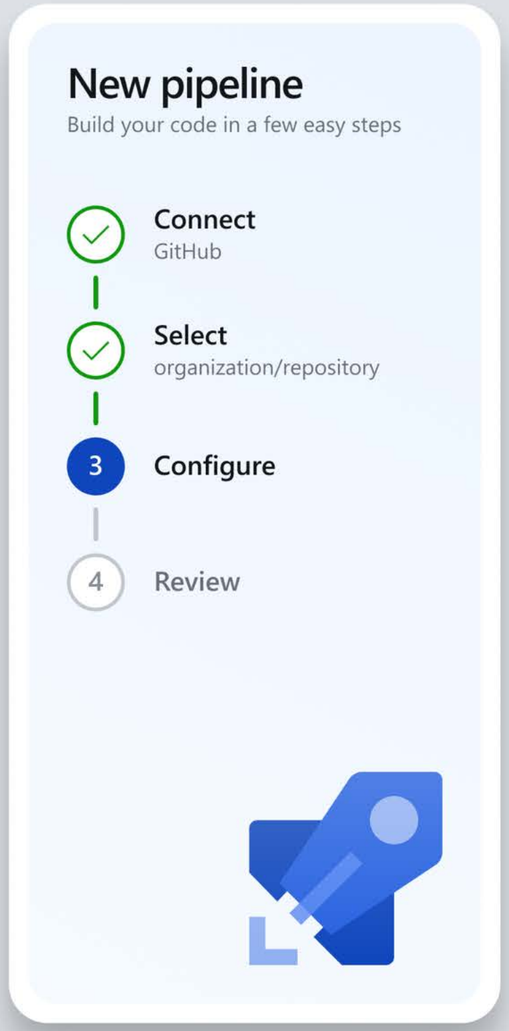
Users who install the GitHub marketplace app get to skip this view.

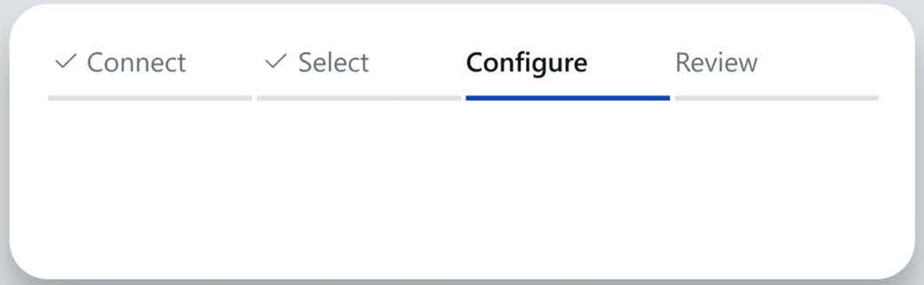
The list of supported version control services continued to grow over time.

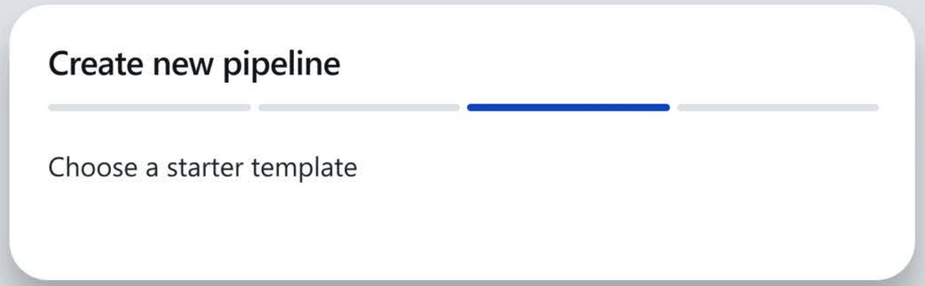


Conveying progress

I did several explorations for the “stepper” component. We decided to keep it pretty minimal to reduce the chances of having it clash with future design system elements.



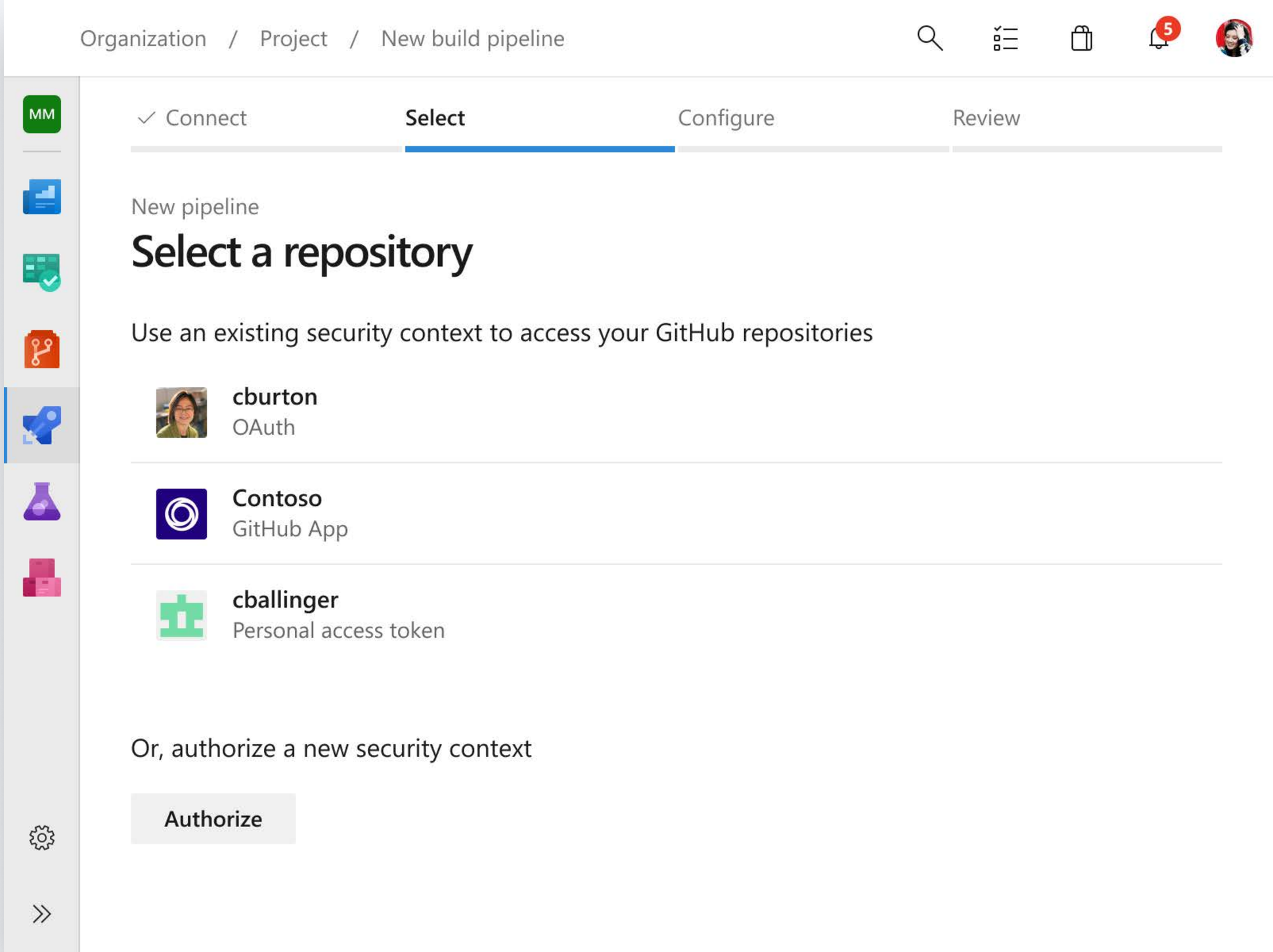




Authorize access to your repository

Users who install the GitHub marketplace app get to skip this view.

For other users, support alternative authentication methods was still required.

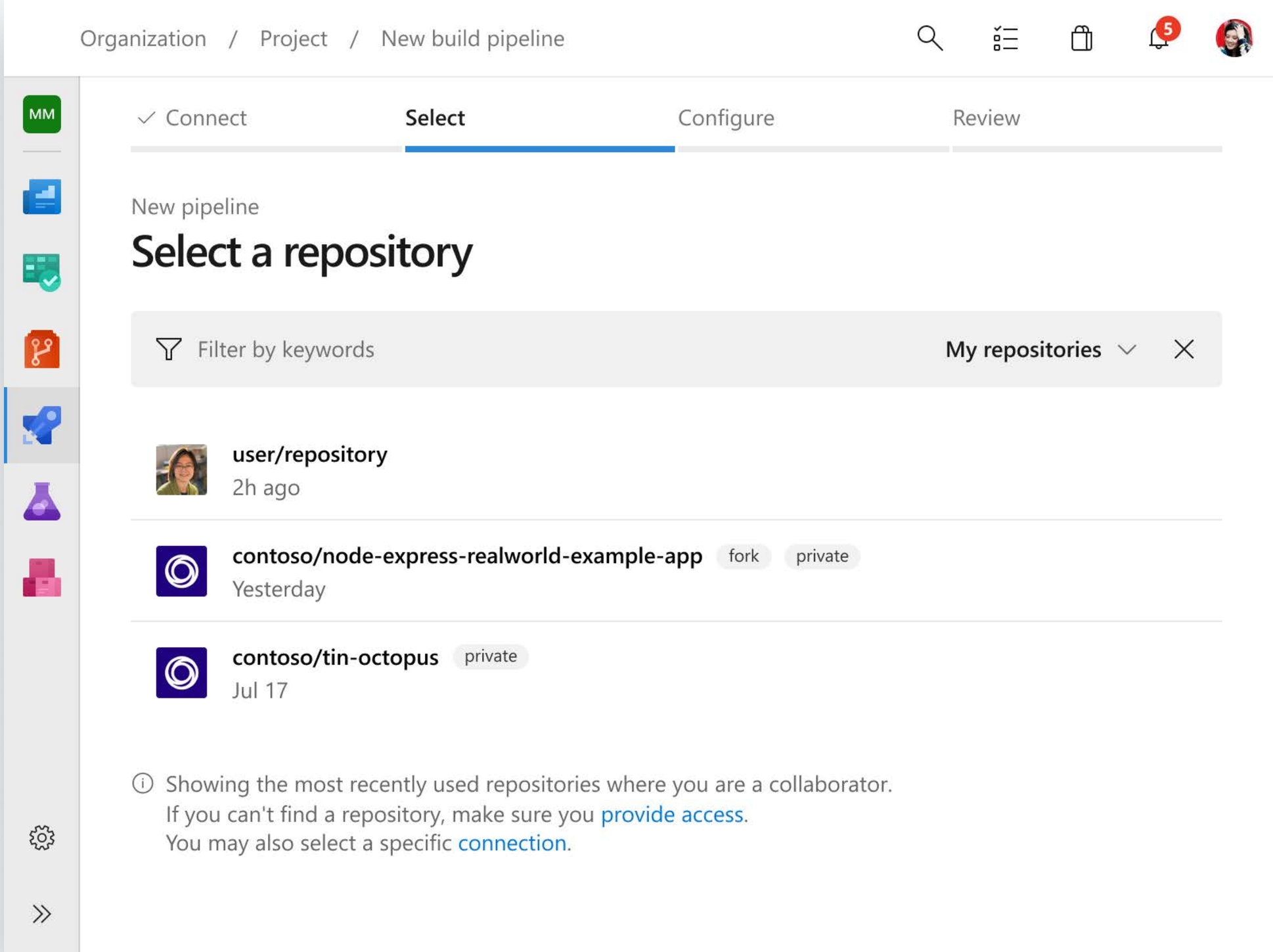


Select your repository

By default, the list includes repositories that you own or collaborate on, sorted by how recently you accessed them.

Annotations call out forks and private repositories for easy differentiation.

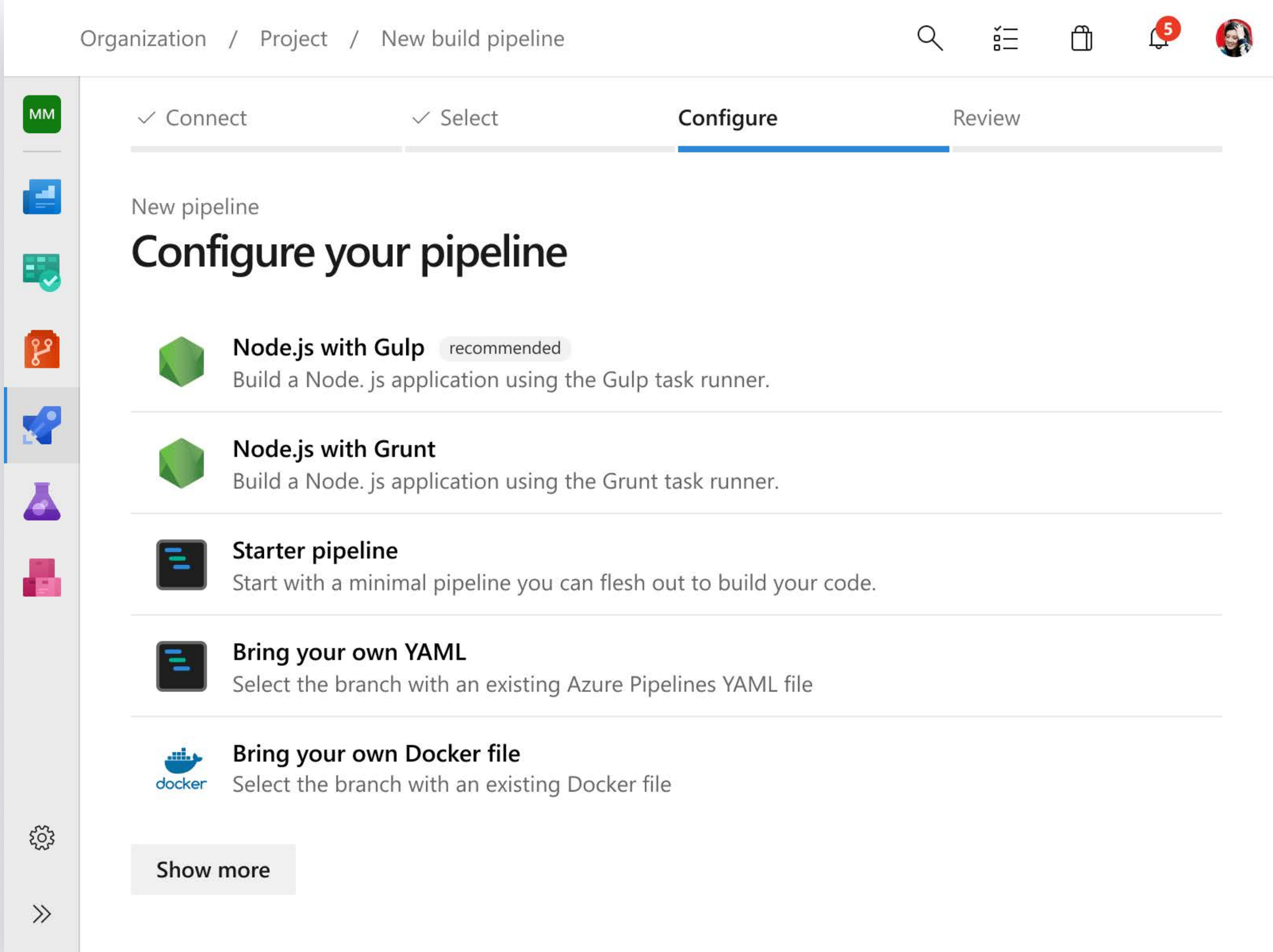
For some users, this list can include thousands of repositories, so we decided to support filtering.



Analyze source code & suggest a starter template

Templates are suggested based on the detection of key files/folders within the repository.

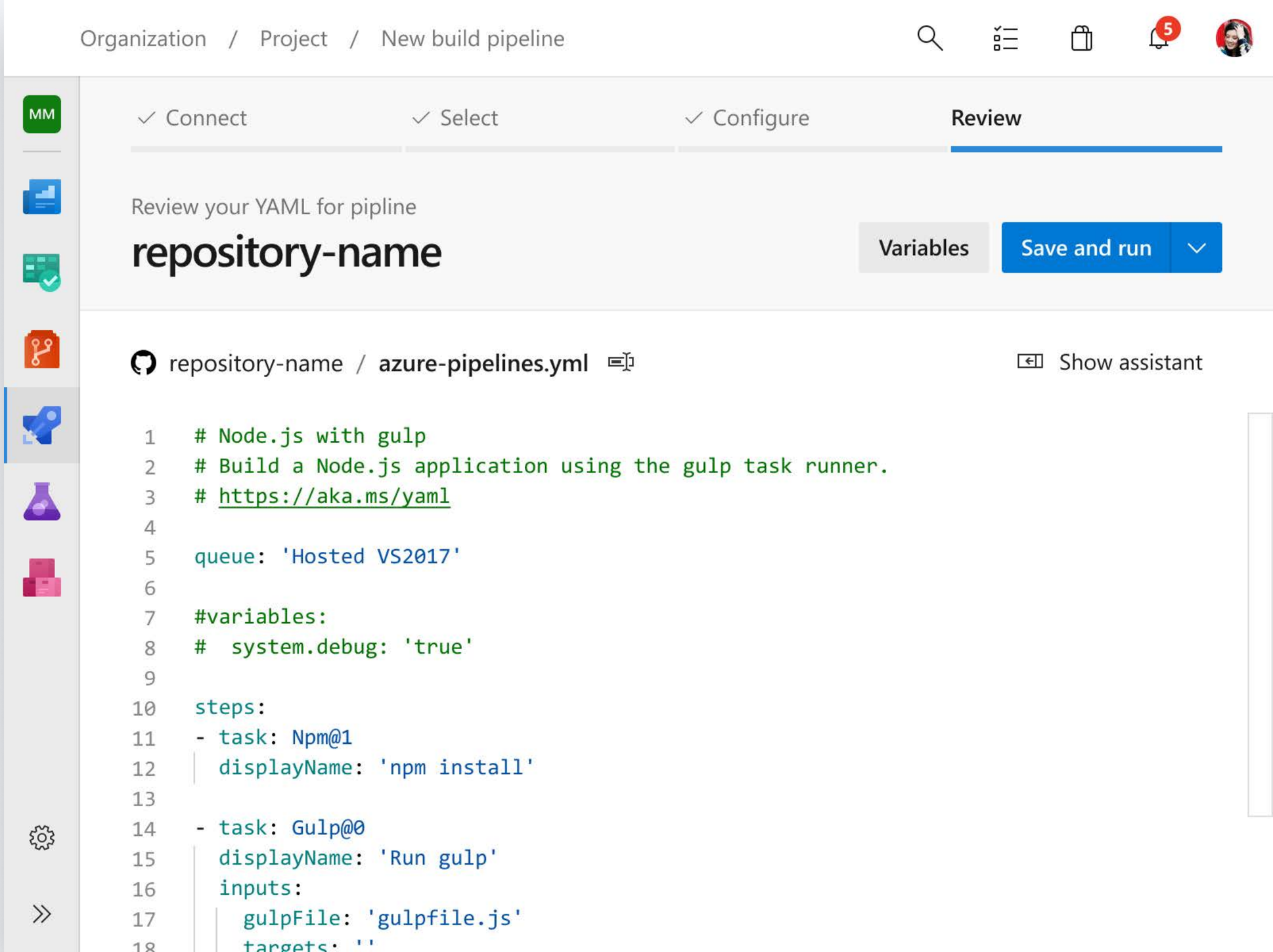
Engineering was skeptical that this analysis would work reliably, but some quick prototyping demonstrated that it would be relatively straightforward.



Generate YAML using the template

While the system can generate the YAML for the pipeline, the user will often want to customize it.

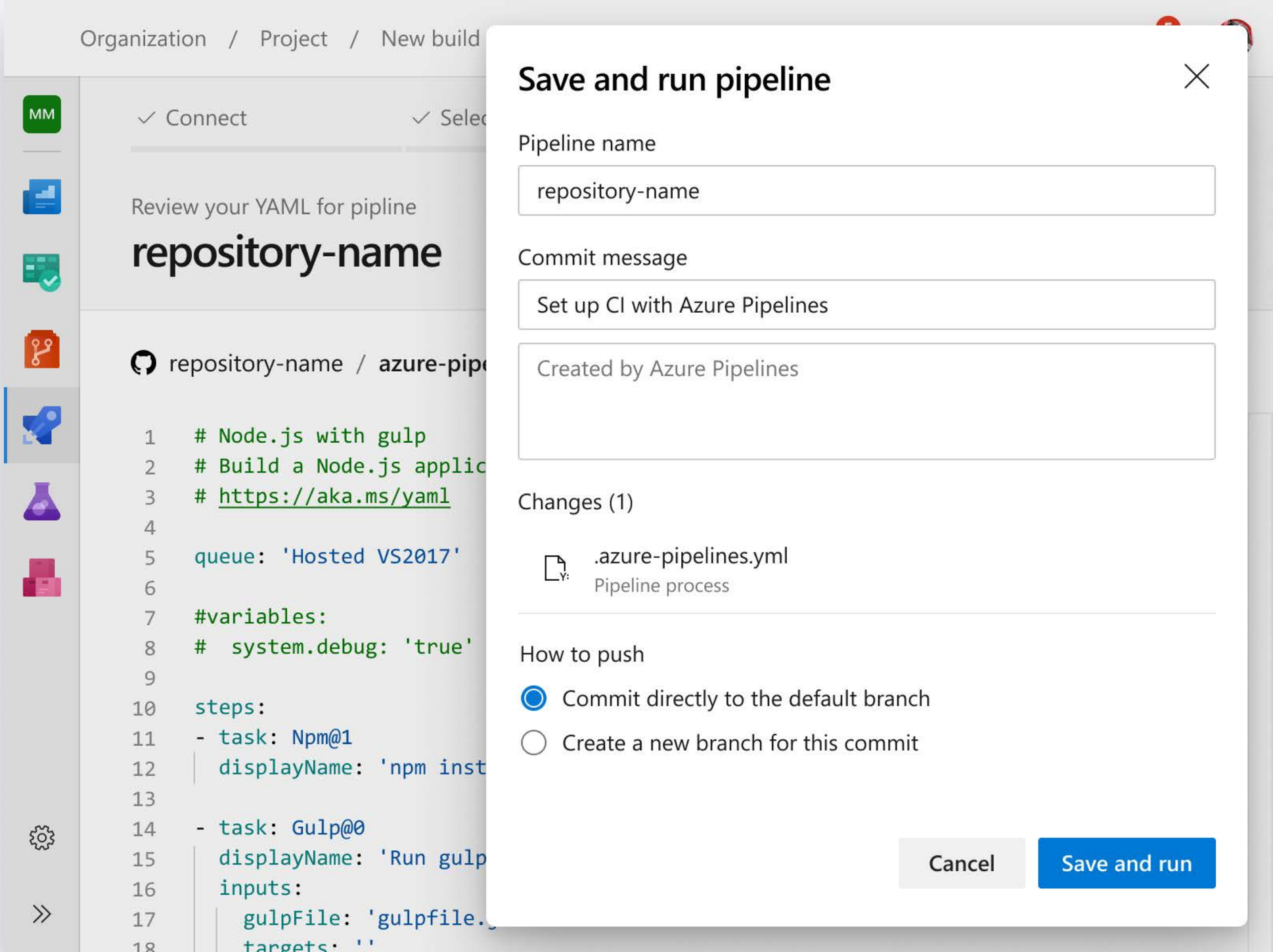
The text editor is full-featured. It uses the Monaco component (used by Visual Studio Code) and includes an assistive panel.



Commit the changes to the repository

Completing the process requires committing the changes to the specified repository.

The panel provides reasonable defaults while providing some flexibility on the details of the commit.



Outcomes

- 🙂 Customer acquisition rates from GitHub increased dramatically.
- 🙂 Added the stepper (progress) component to our nascent design system.
- 🙂 Engineering was able to reuse the experience for all of the pipeline creation flows.

Lessons

Cross-service authentication is hard, but GitHub's new app model helped us simplify it dramatically.

Balance approachability for new users with flexibility for experienced ones.

When getting ahead of the design system, partner with avoid creating future design debt.

The Challenge

- ✔ Integrate the build and release areas of Azure DevOps into a single pipeline experience.
- ✔ Support configuration as code.
- ✔ Preserve support for complex scenarios while simplifying the experience for most users.

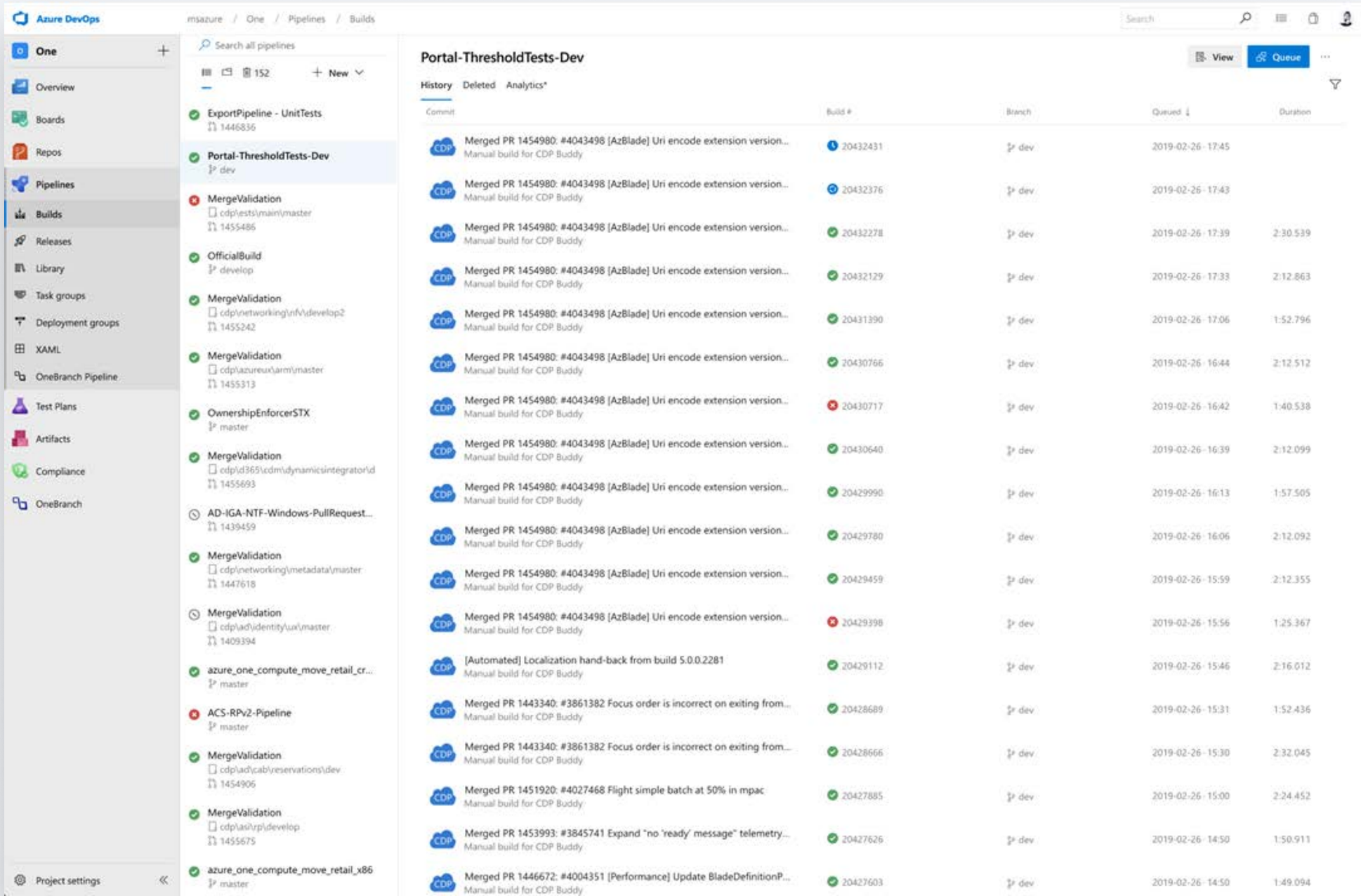
My Role

I worked with two other designers (who were new to the team) and two researchers for over nine months.

I was the sole designer for the second iteration of this design. The goal was to address the user feedback from the preview of the initial design.

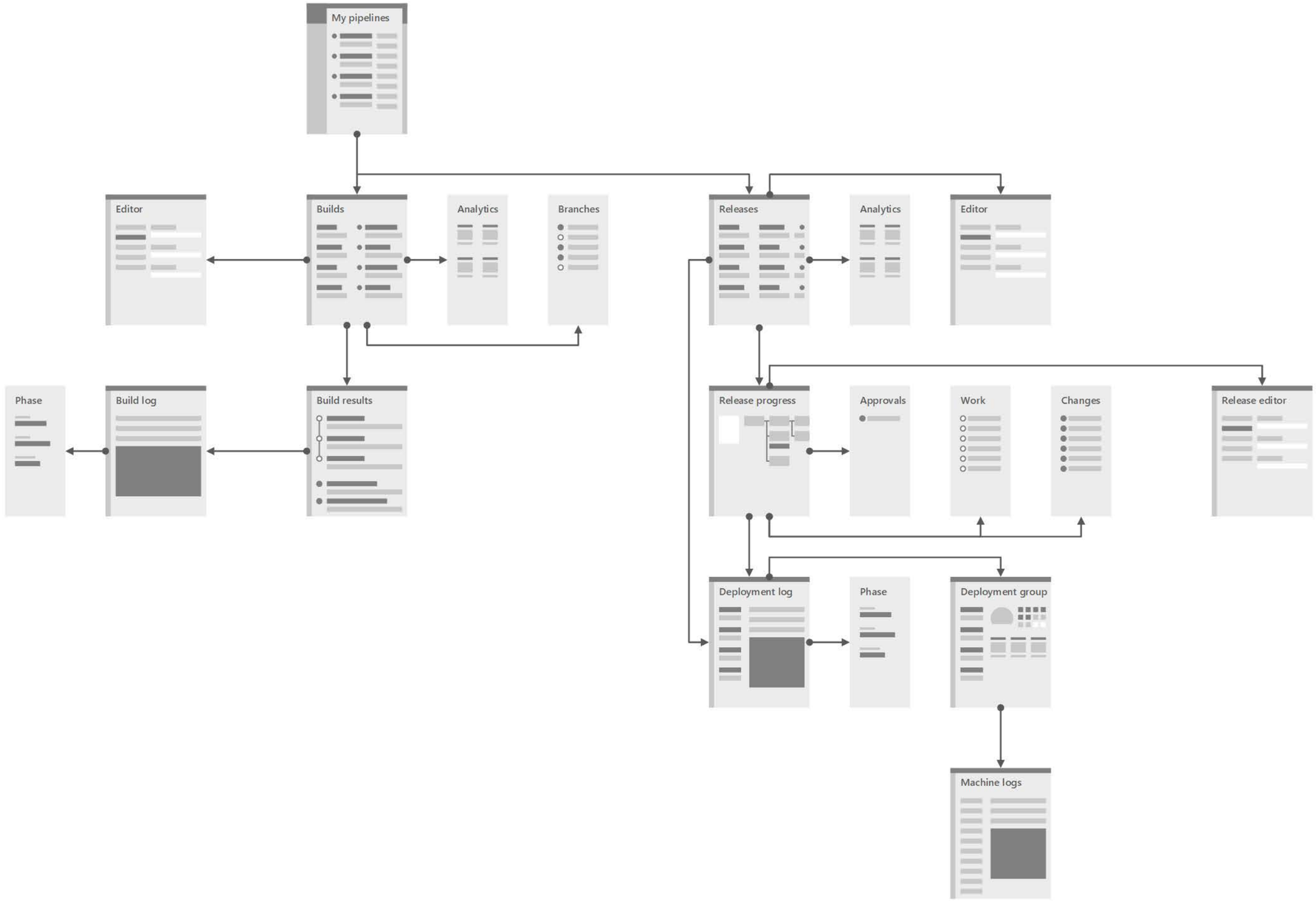
Existing UI

The views for automated builds were quite complex and could be challenging to navigate. This particular screenshot represents a “worst-case” scenario, but you can see how it can be overwhelming.



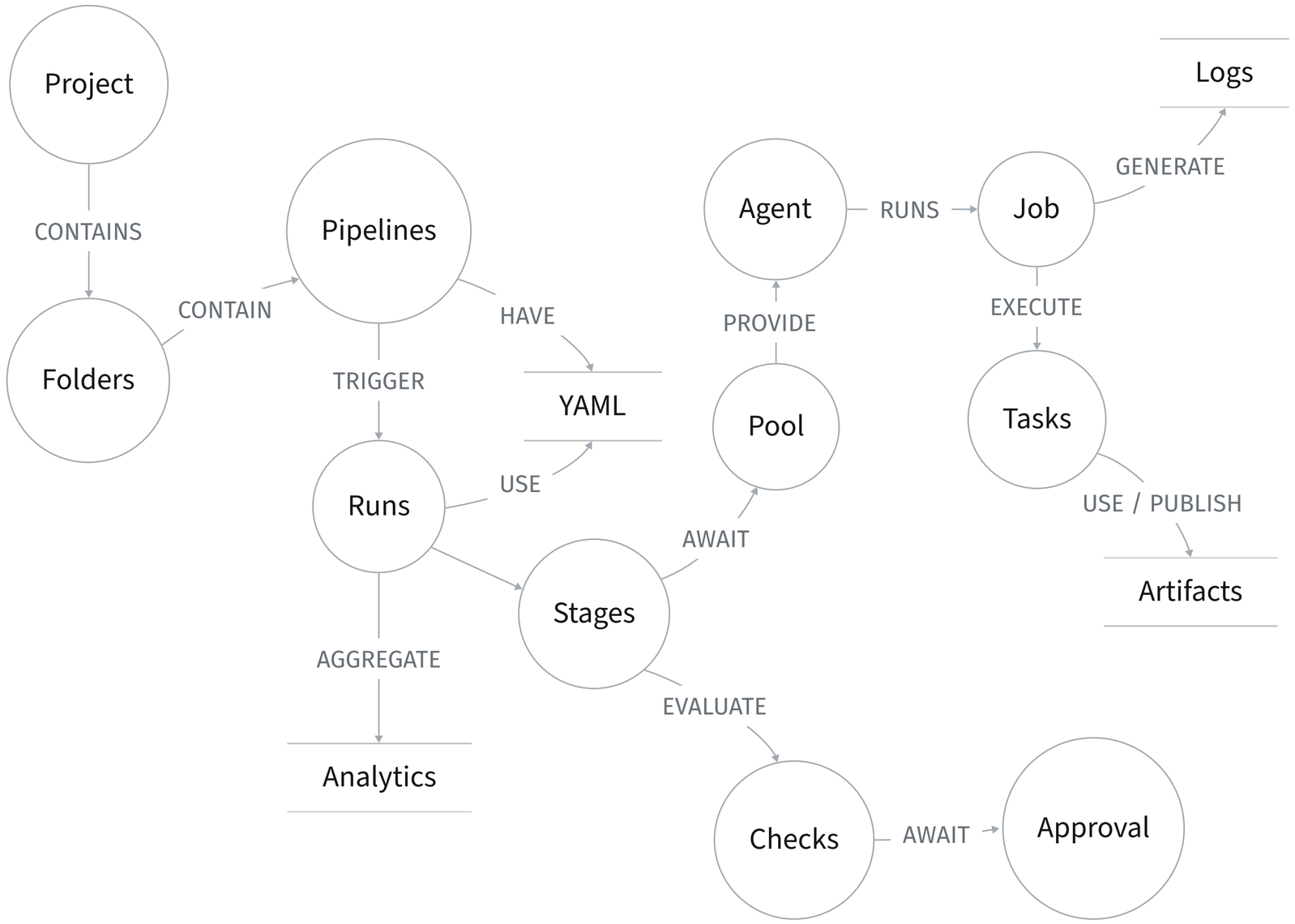
Wireflow

I created a wireflow (https://www.nngroup.com/articles/wireflows/) to map out how the various views related to one another.



Information architecture

I created this more conceptual diagram to map out the various objects within the system and their relationships.

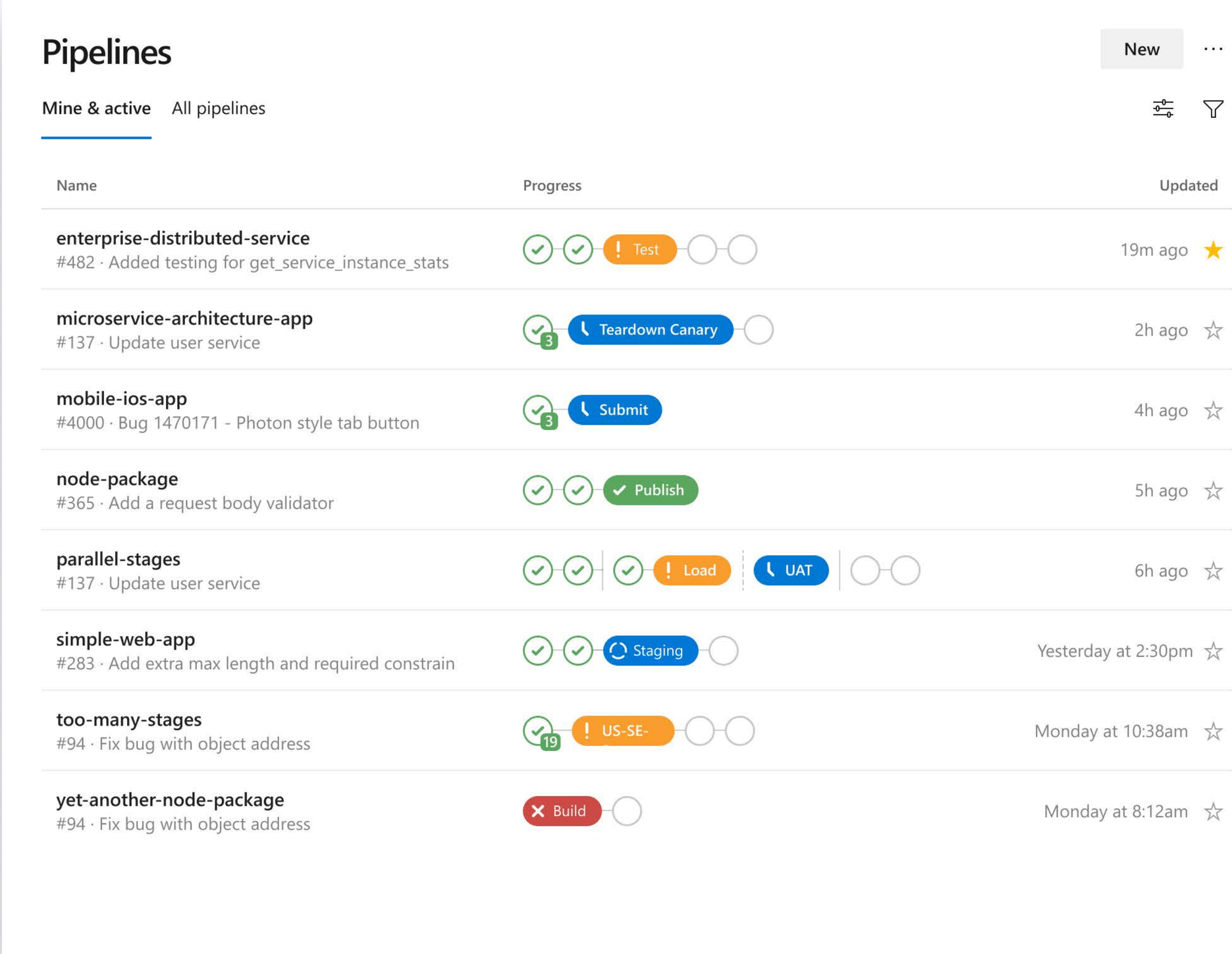


Early exploration

In this exploration, I pushed the limits of how much information I could represent in the pipeline diagram.

I put a heavy visual emphasis on the most recently completed stage and de-emphasized stages that finished successfully.

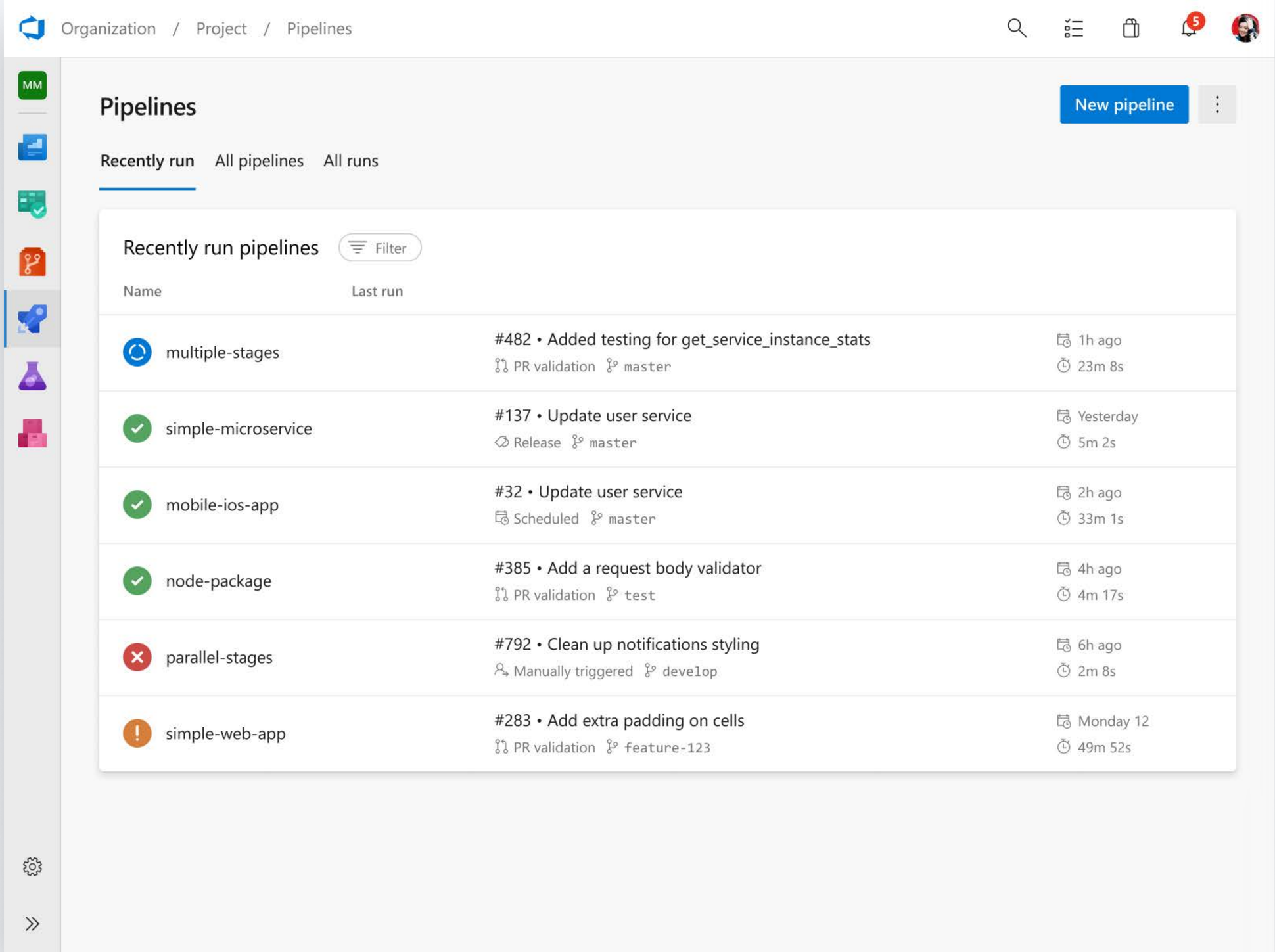
I used vertical delimiters to represent stages running in parallel.



Listing

The initial preview only supported single stage pipelines, so we went with a simpler visualization of pipeline status.

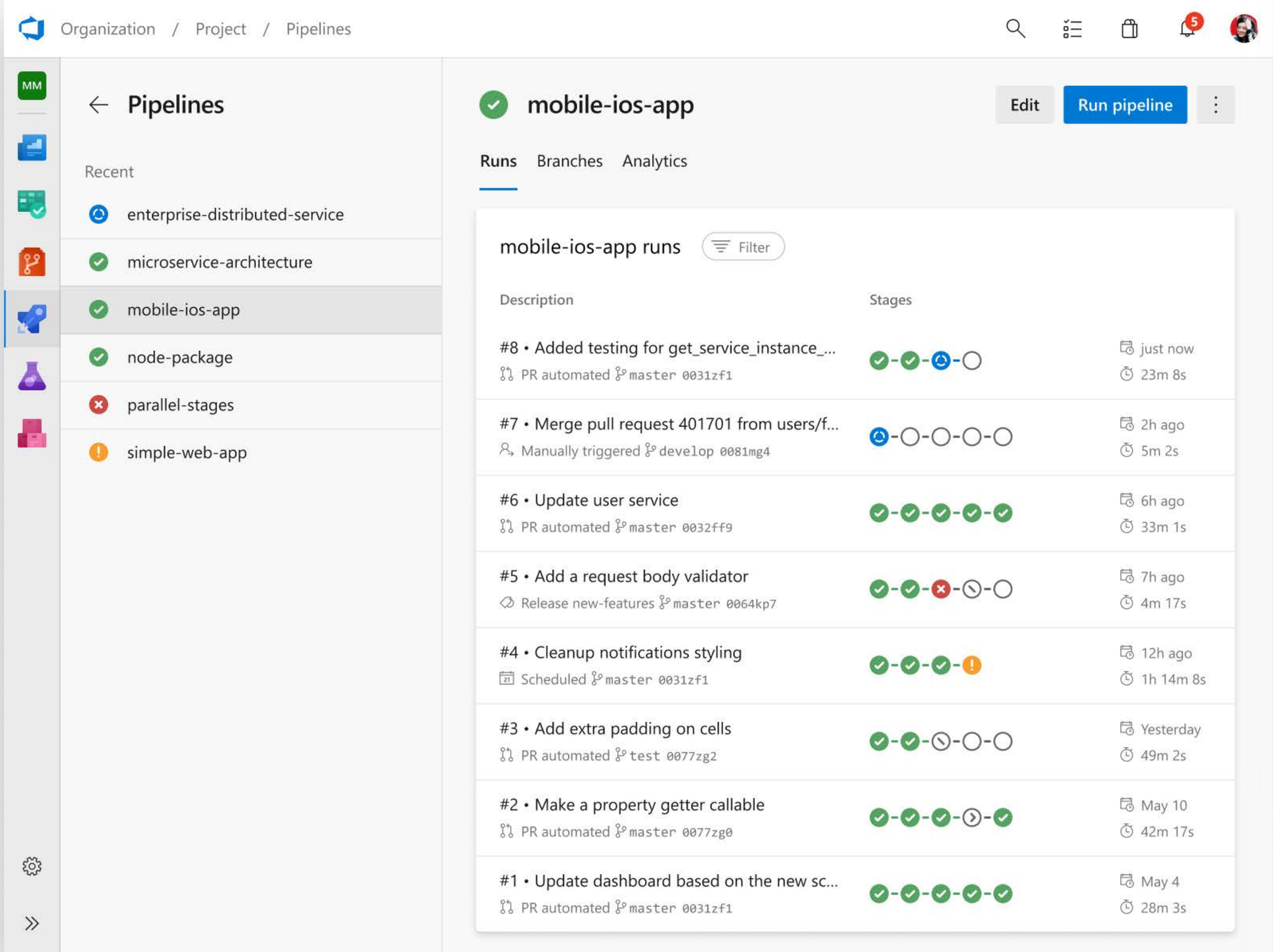
Each row has a default click target as well as sub-targets for key elements and contextual commanding.



Detail, runs

Selecting a pipeline transitions to this view. A simplified version of the pipeline list appears on the left. The right-hand view shows the runs for the selected pipeline.

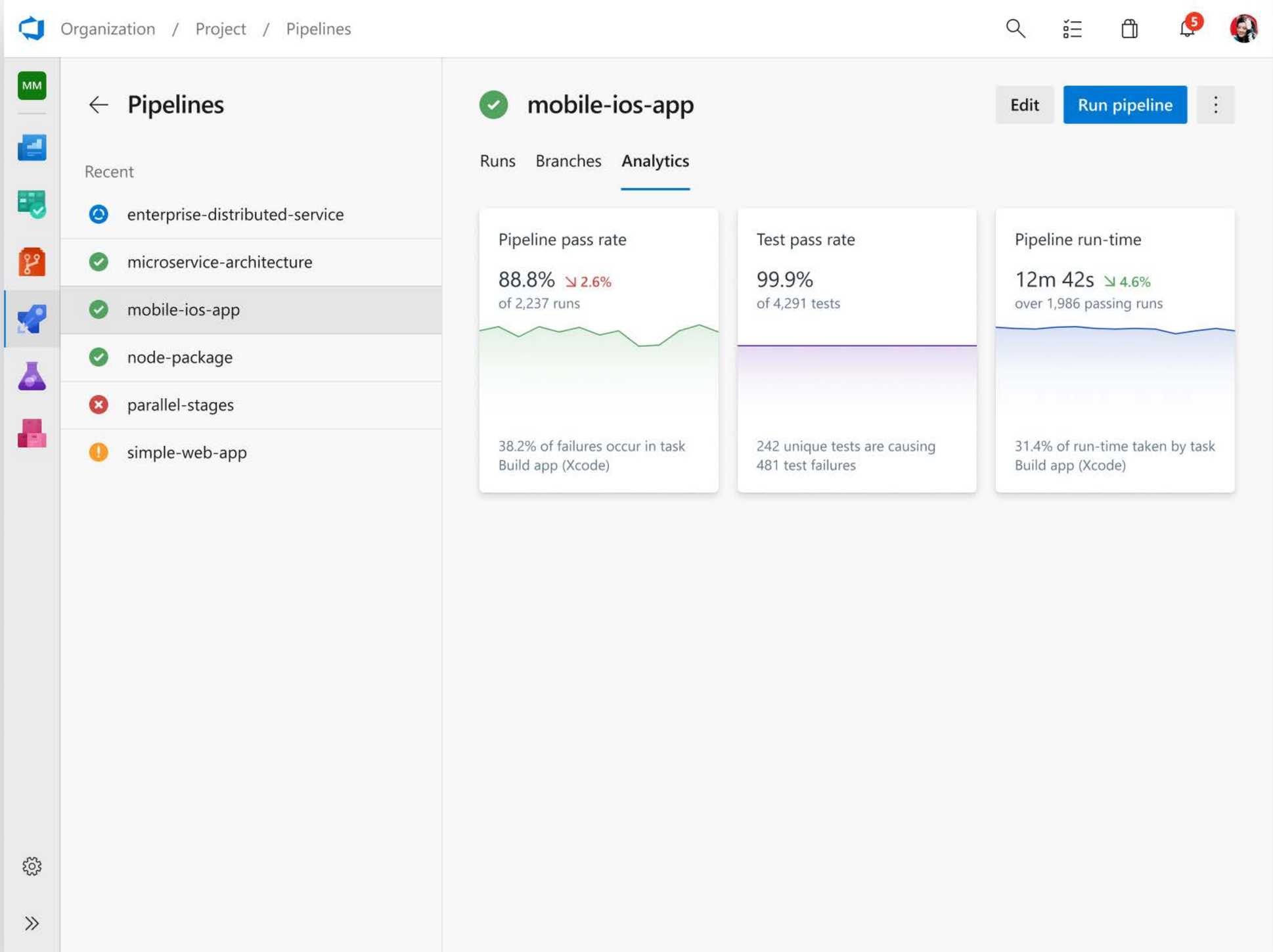
The transition helps keep the user oriented and makes it easy to check up on the status of multiple pipelines.



Analytics

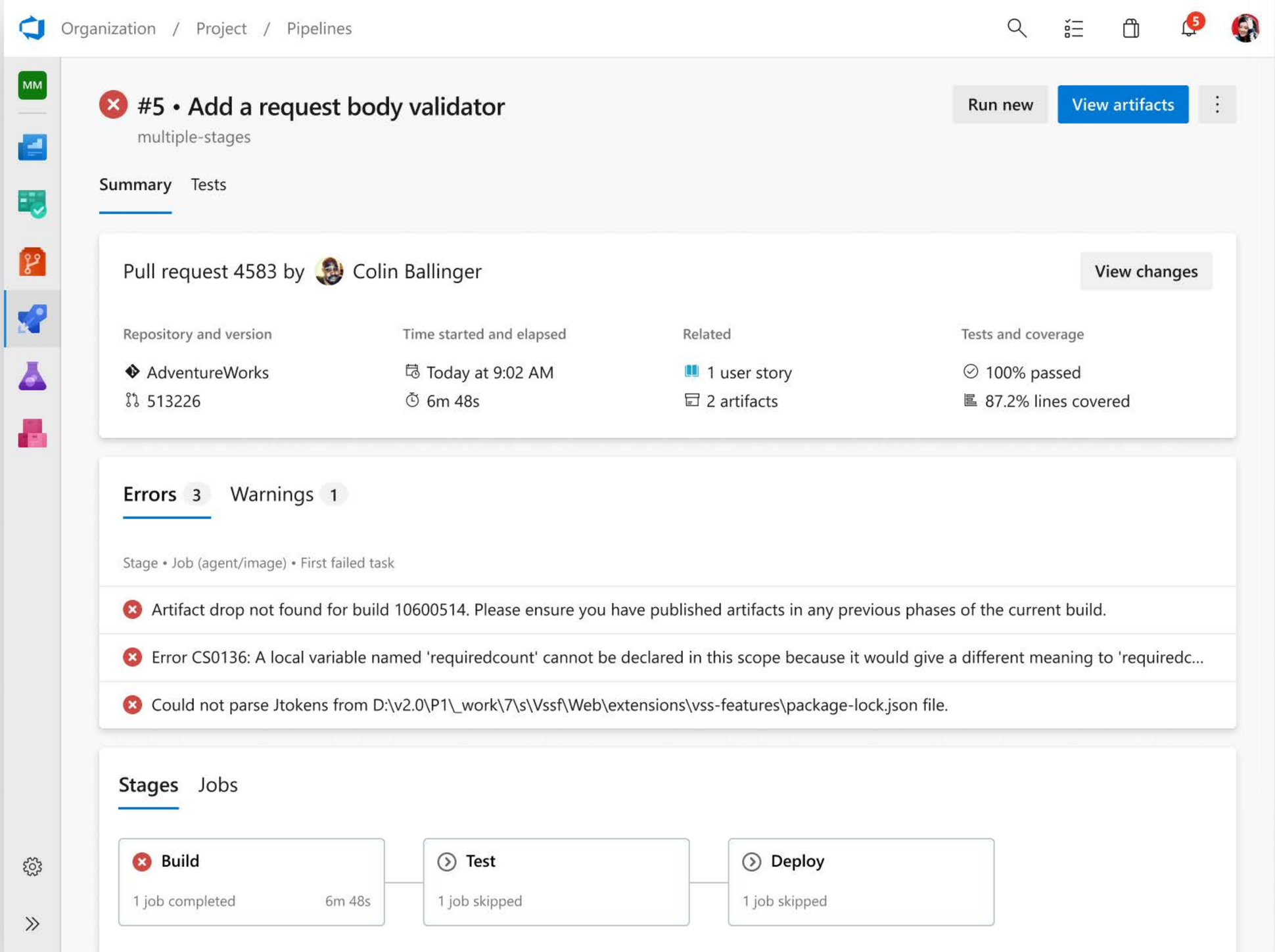
The analytics tab shows some key performance indicators for the selected pipeline.

The note at the bottom provides some analysis of the data beyond the graph.



Error presentation

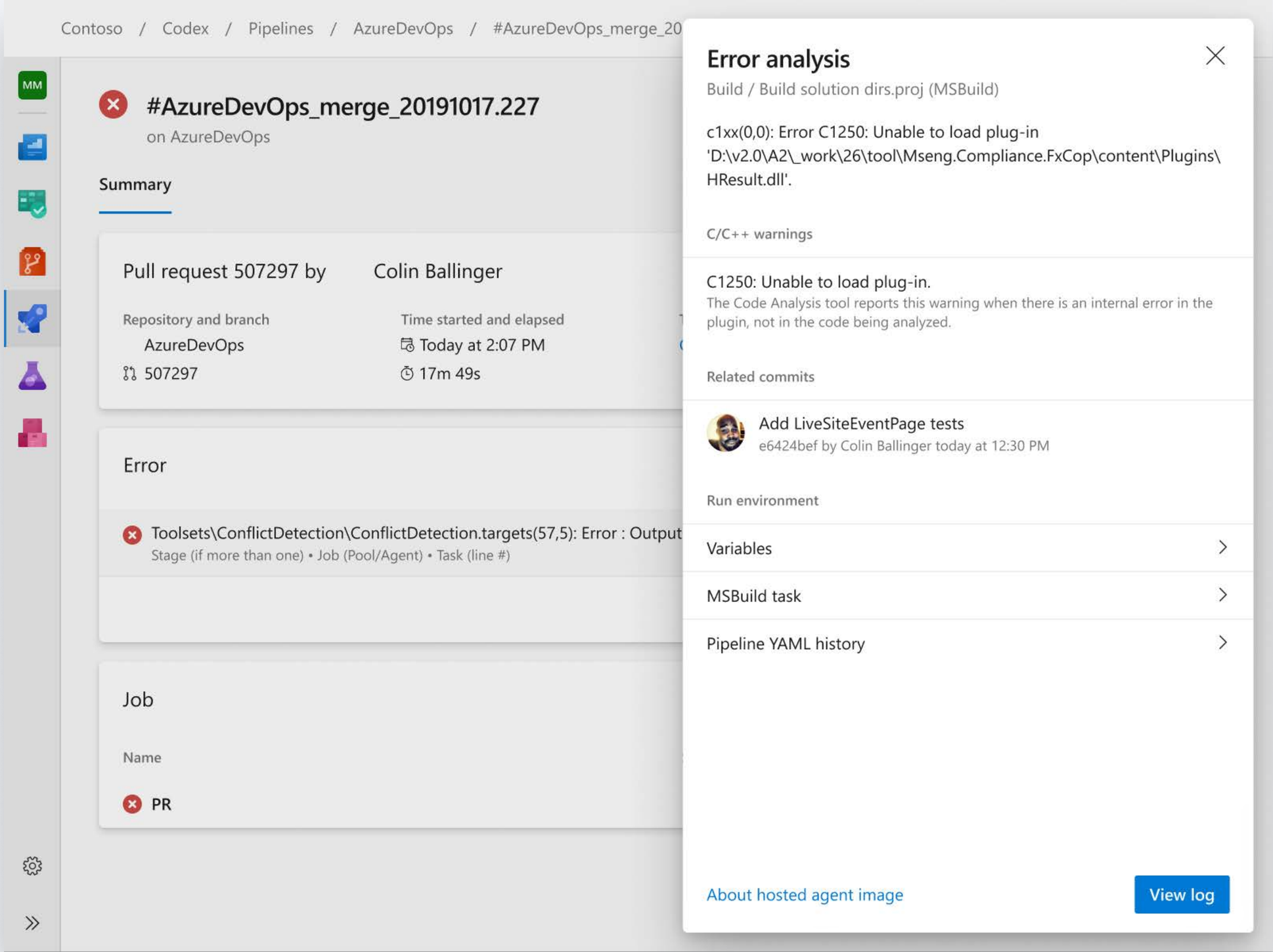
When a run fails, we want to make it as easy as possible to review and diagnose any errors or warnings.



Failure analysis

I explored the idea of having an assistive experience to help users diagnose problems. My goal was to reduce support costs and improve user satisfaction.

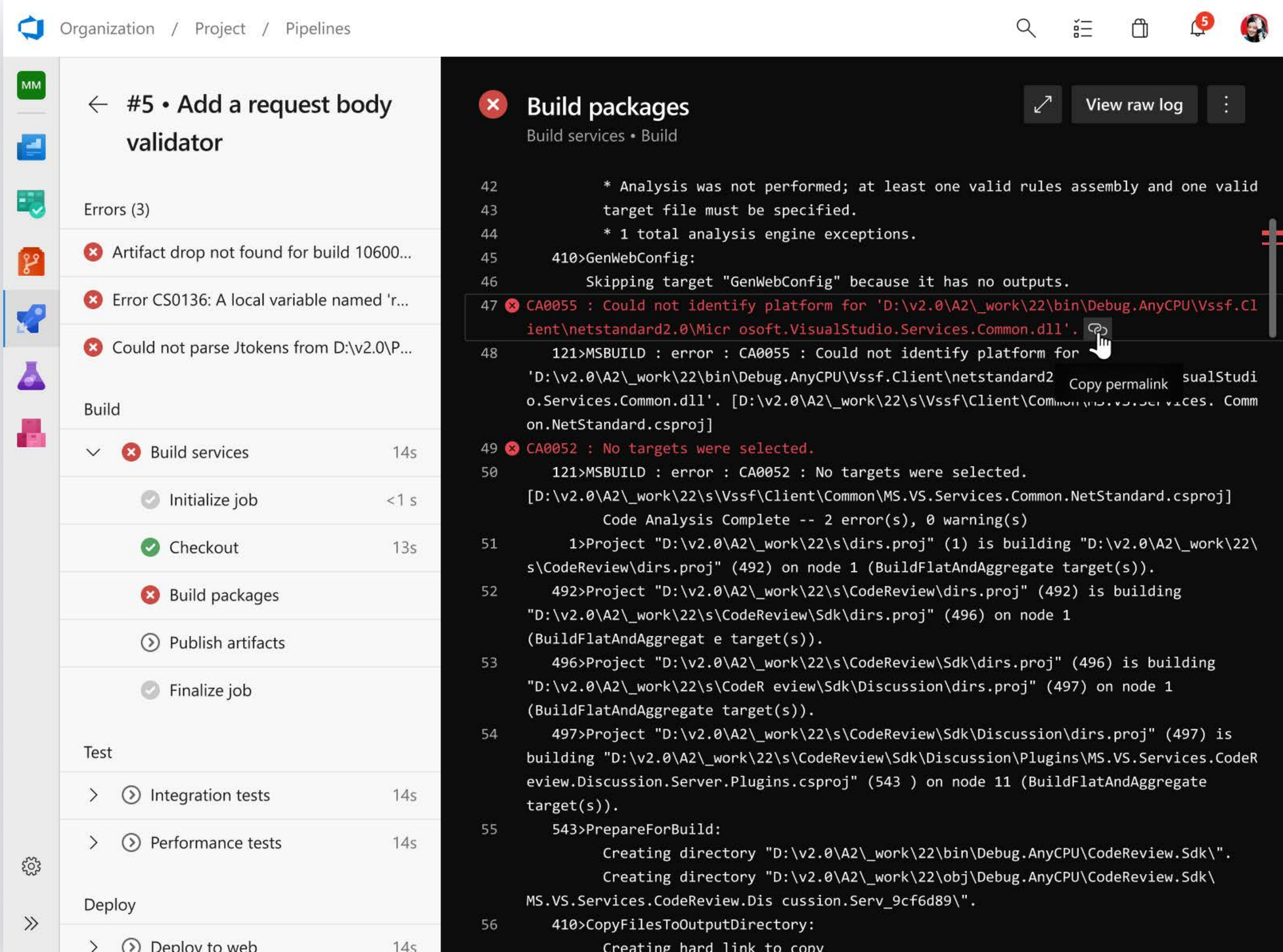
This feature didn't make it into the initial feature preview or the general availability release. Still, product management put it on the backlog, and engineering built it out a few months later.



Logs

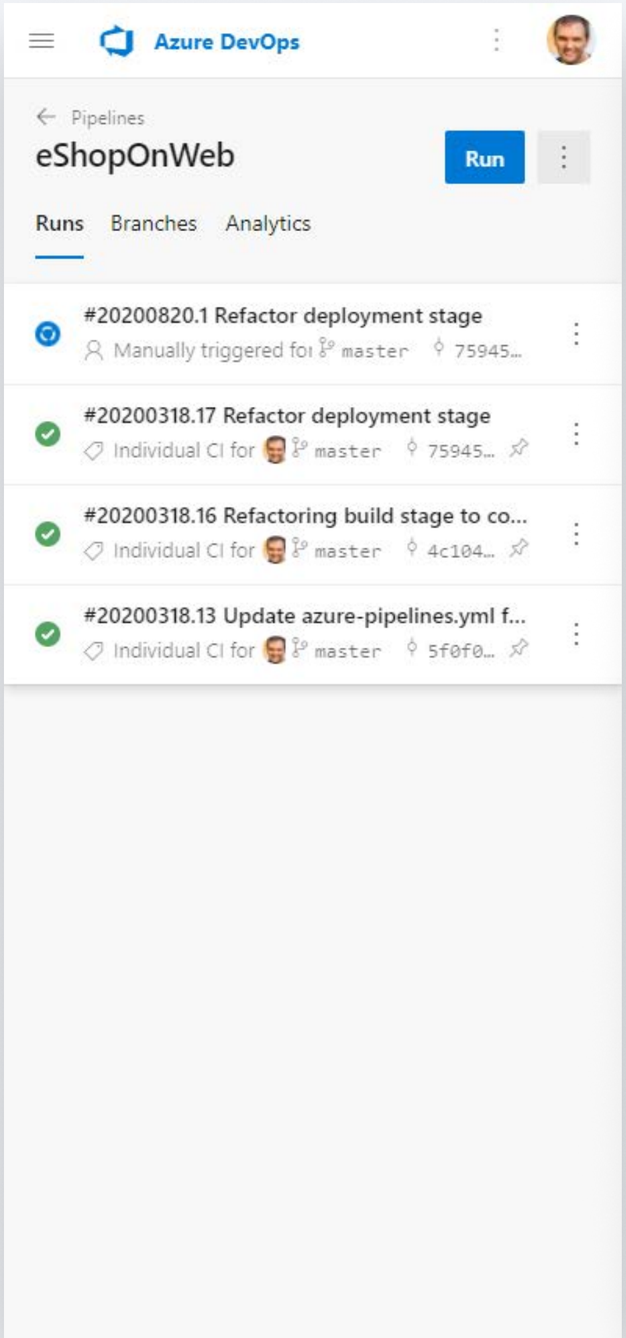
The logs view has to handle tasks in a range of states (pending, in progress, and completed). Each state has a unique set of attributes.

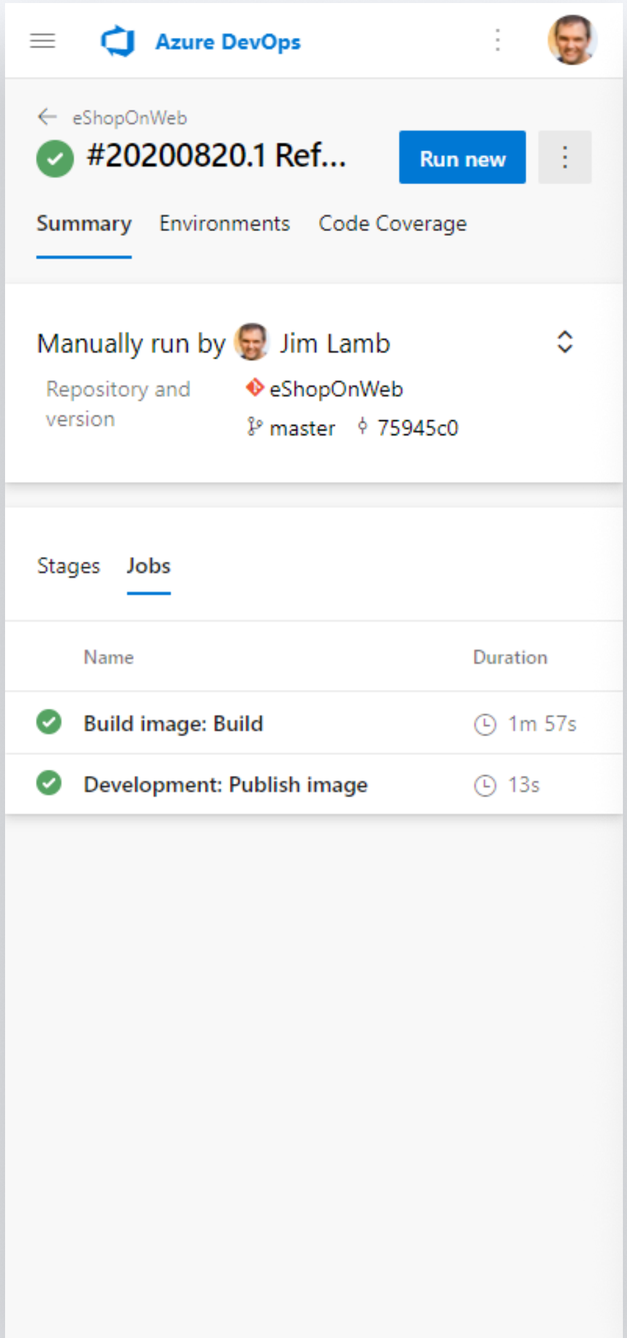
This view also has to scale to very, very large logs while being highly performant.

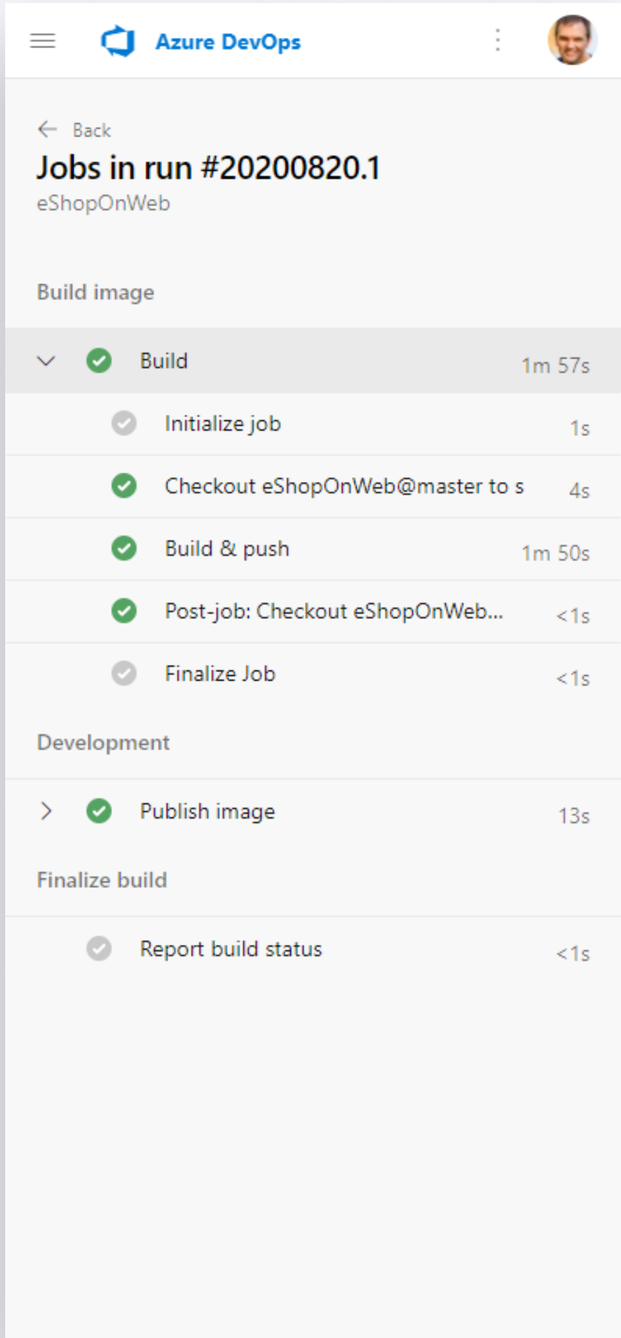


Mobile views

We also designed mobile (responsive) versions of commonly used views.







Outcomes

- 🙁 Initially high opt-out rates due to navigational changes.
- 🙂 Dramatically reduced opt-out rate during feature preview by addressing user feedback.
- 🙂 Improved adoption of single pipeline solution over legacy features.

Lessons

Simpler isn’t always better.

Design for scale *and* performance.

Don’t over-index on features users haven’t adopted yet.

Other recent projects

- ✓ Improvements to the logs view for pipeline runs
- ✓ Onboarding to Azure's Kubernetes Service
- ✓ Notifications for Teams & Slack
- ✓ Pipeline YAML editor with integrated assistance
- ✓ Accessibility Insights integration with GitHub actions

Design system contributions

- Navigating multiple pages within a panel/fly-out
- Empty states
- Master/detail views
- Accordion, collapsible cards
- Segmented control (choice chips)
- Multi-pane workspaces

Thank you!

Any questions?

You can find me at:

https://www.linkedin.com/in/jlamb

jimlamb@hey.com