**Exercise**

**Monitor the health of your pipeline**

In this exercise, you examine a few of the analytics features that Azure Pipelines provides.

Irwin asked the Tailspin team how they can release faster. Building an automated release pipeline is a great step toward releasing quickly and reliably. As you release more often and more rapidly, it's important to understand the health and history of your releases. Looking at health trends regularly can help you diagnose potential problems before they become critical.

Before we take a look at some of your pipeline's analytics, let's listen in on the Tailspin team at their morning meeting.

**How can I track the health of my entire pipeline?**

It's the following morning. At the team meeting, Andy and Mara have just finished demonstrating the build and release pipeline that they set up as a POC.

**Amita:** This POC is fantastic. The build pipeline was a great start, but I still had to manually install the build artifact in my lab so I could test it. If I can get these releases to my test environment on a regular schedule, I can move new features through QA much faster.

**Mara:** Exactly! And remember, we can always expand our release pipeline to include more stages. The goal is to create a complete deployment workflow.

**Tim:** A staging environment would be great. I could do additional stress testing before we present new features to management for final approval.

*The team is excited to see what the new pipeline can do. They all start talking at the same time.*

**Andy:** I'm excited too. But let's take one step at a time. Yes, I think we can make all of these changes and more, but this is just a POC. We'll work on expanding it over time.

**Amita:** A significant question for me is, how do we know if what we have in this environment is valid? I mean, where do I go to see the health of the releases?

**Andy:** Remember the dashboard we created to monitor the health of the builds? We can set up the same kind of system for the releases. We'll be able to monitor the health of the whole pipeline. Anyone who's interested can check in on the system and see what's happening.

**Tim:** Irwin will like that.

**Andy:** Let's hold off on building a release dashboard until after we've defined a more complete release workflow. For now, let's look at some of the built-in analytics that Azure Pipelines provides.

The team gathers around Andy's laptop.

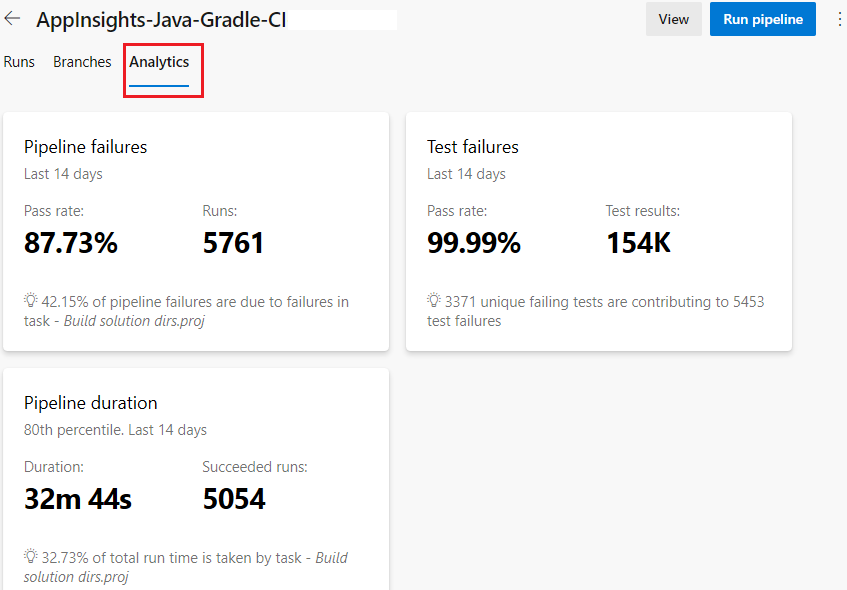
**What information does pipeline analytics provide?**

Every pipeline provides reports that include metrics, trends, and insights. These reports can help you improve the efficiency of your pipeline.

Reports include:

* The overall pass rate of your pipeline.
* The pass rate of any tests that are run in the pipeline.
* The average duration of your pipeline runs, including the build tasks, which take the most time to complete.

You'll examine the analytics for your pipeline shortly. Here's a sample report that shows the pass rate and duration for a project that has many pipelines runs.



You can filter the results to focus on a time range or the activity on a Git branch.

Azure DevOps also provides this information as an OData feed. Use this feed to publish reports and notifications to systems such as Power BI, Microsoft Teams, or Slack. You can learn more about analytics feeds at the end of this module.

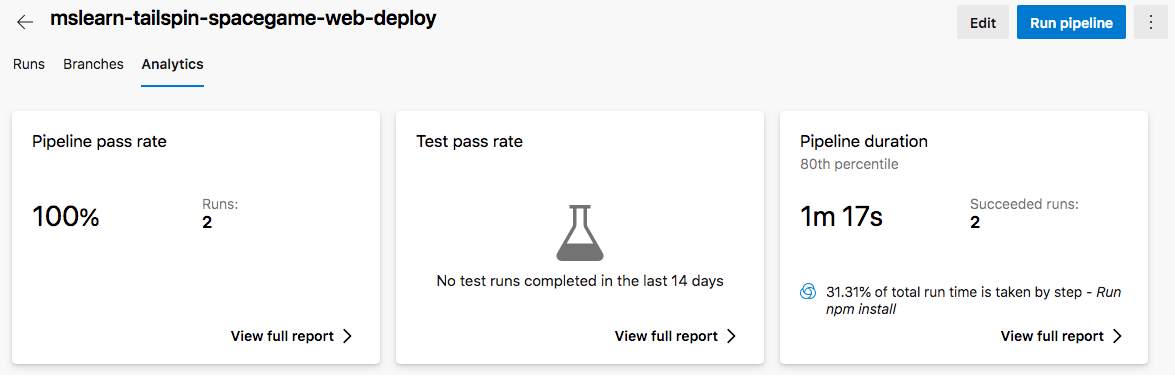
**Explore your pipeline's analytics**

Let's examine a few of your pipeline's analytics.

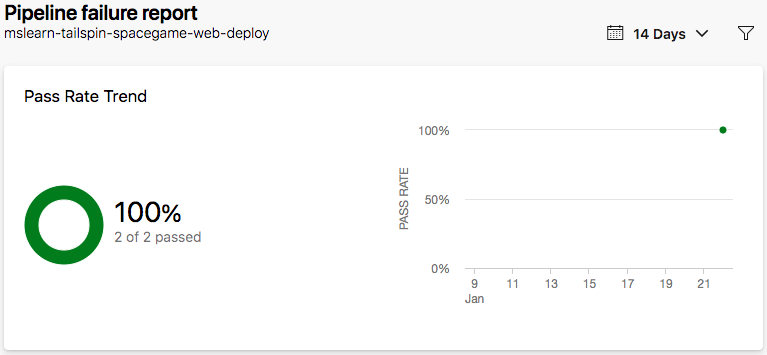
1. In Azure Pipelines, navigate to your pipeline.
2. Select the **Analytics** tab.

Azure Pipelines, showing the Analytics tab

1. Note the pass rates and average duration of your pipeline runs.



1. Under **Pipeline pass rate**, select **View full report**.



At this point, you have a basic pipeline report that contains data for just a few runs.

**Amita:** That's the information I want, but I don't see much data yet.

**Andy:** That's right. We'll collect more data as we perform more runs over time. We'll use this data to gain a greater sense of the health of the pipeline and where we can make it more efficient.

**Mara:** For example, I see that the task that runs npm install takes the most time to finish. Do we need to fix that? Perhaps we can make it run faster by caching the Node.js packages that we use on one of our servers.

**Andy:** I always appreciate your willingness to improve things. But let's hold off on that idea for now. Time will reveal where the pipeline is slow or inefficient. From there, we can decide which improvements are worth tackling.

# Plan the tasks for the upcoming sprint

* 2 minutes

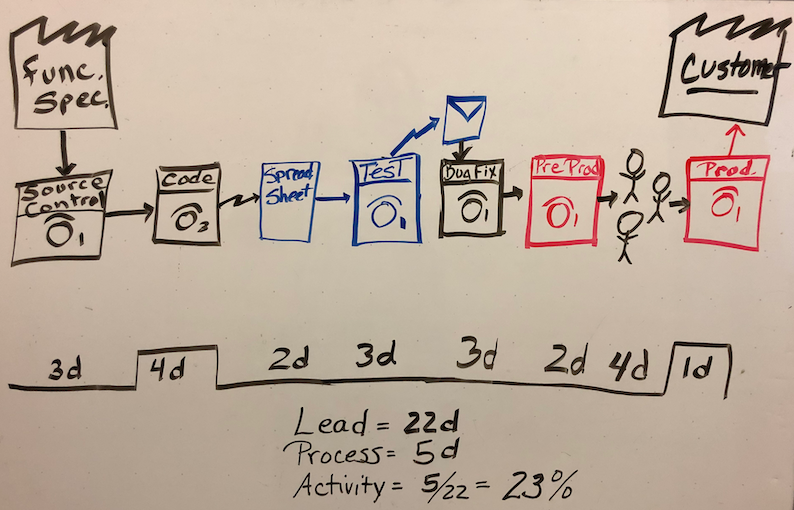
With a successful POC and some basic monitoring in place, the team is once again excited to see what this pipeline can do. Tim and Amita have questions for Andy and Mara.

Andy glances at Mara and clears his throat.

**Andy:** Hold on. Let's step back and think about this setup in terms of our new Agile approach. We finished our last sprint by accomplishing two goals. We created our backlog by using Azure Boards, and we created our pipeline by using Azure Pipelines. Now we need to set up our next sprint iteration. What's our goal, and what are our tasks to accomplish it?

**Amita:** Andy is right. Irwin asked us to find a way to speed up deployments. I think that should be the focus of the next sprint. Remember our value stream map?

Amita points to the value stream map, which you originally saw in the Assess your existing software development process  module.



**Amita:** In several areas, we can improve by cutting down the time we spend. For example, we're still slow at running quality tests.

**Tim:** Exactly. And we still have a bottleneck at deploying changes to production because management needs to approve new features before we release them.

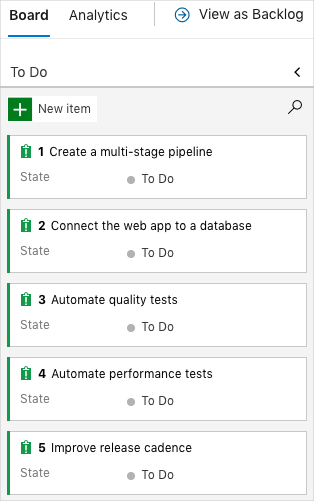
**Mara:** It sounds like we can start planning the tasks we'll accomplish in this sprint.

The team brainstorms and comes up with these tasks:

* Create a multistage pipeline.
* Connect the web app to a database.
* Automate quality tests.
* Automate performance tests.
* Improve release cadence.

**Mara:** If these tasks seem right, I'll add them to Azure Boards.

The team agrees. Mara sets up a sprint iteration and adds these tasks:



Each of these tasks relates to what you'll do in the following modules. You don't need to add the tasks to Azure Boards now. You'll run a template that populates Azure Boards with these tasks, just as you did in previous modules.