

eBook

Finding Build Failures

Investigating build failures with CloudBees Pipeline Explorer

Ever found yourself in these kinds of situations? Let me paint a picture for you.

Imagine it's Friday, day five of the sprint. You're feeling good because you just wrapped up testing a cool new feature right before lunch. You've also tackled all the review comments, pushed your changes, and now you're letting the CI build run its checks while you grab a bite. Sounds pretty smooth, right? But then, after lunch, boom! The build fails.

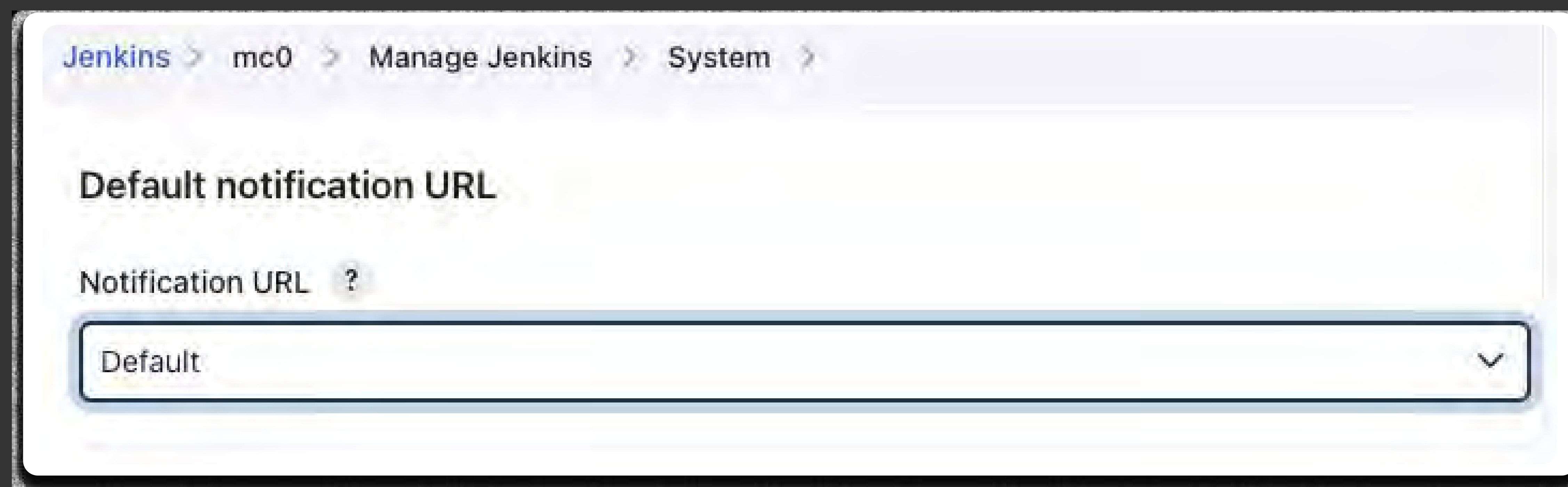
Or how about this one?

You're on GitHub, staring at all these PRs. You're ready to dive in, make some changes, merge the code – the usual drill. But then, your eyes catch those pesky red X's mixed in with green checks. And you're left wondering, "Why on earth is this one marked with a red X?"

In both scenarios, the ball's in your court to play detective – to dig in and figure out what went awry in the execution of a Pipeline, and fast. And if you're stuck, you've got to piece together a story, leaving a trail of breadcrumbs for someone else to follow and help you out. This typically means diving into the pipeline log, downloading it, and then opening it up in Visual Studio Code. You find yourself scrolling through heaps of log lines, hunting for keywords like "failed" and "error". It's a lot of wading through information, looking for that needle in a haystack.

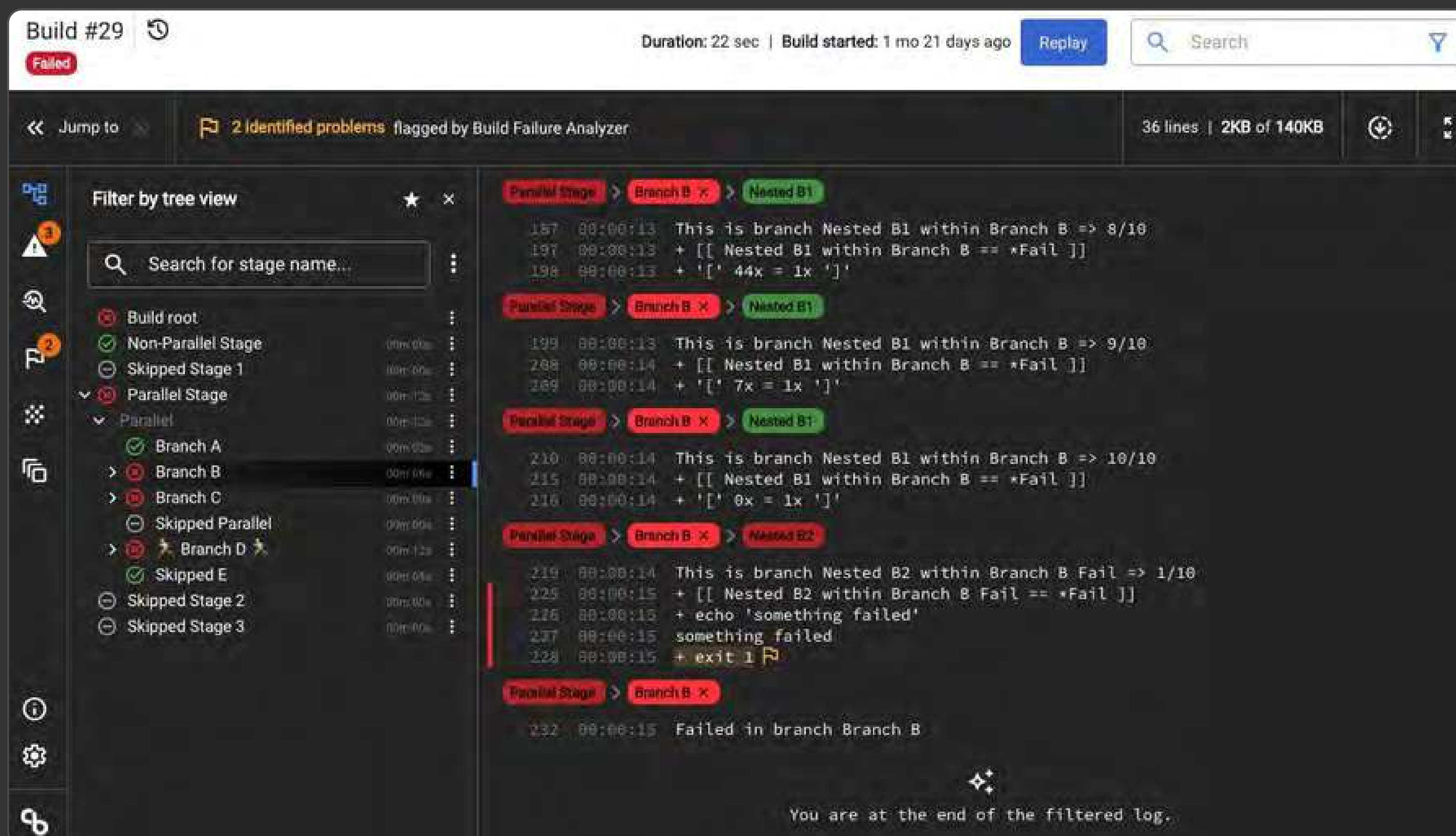
Let's be real, that's a ton of legwork, especially when you've got a whole list of features to deliver. Debugging a pipeline shouldn't feel like you're trying to solve a mystery every single time, especially when it comes to just pinpointing where the error started. So, how about we simplify things? Let's dive into troubleshooting these headaches with the new CloudBees Pipeline Explorer. It's about making your life easier, and I'm here to guide you through it, step by step.

NOTE: Jenkins will use what is in this Default notification URL setting to figure out what screen to take you to for pipeline troubleshooting. If CloudBees Pipeline Explorer Plugin is installed and enabled, the default URL will point to the Explorer. So if you see a different screen - check your Default notification URL in your Manage Jenkins->System view:



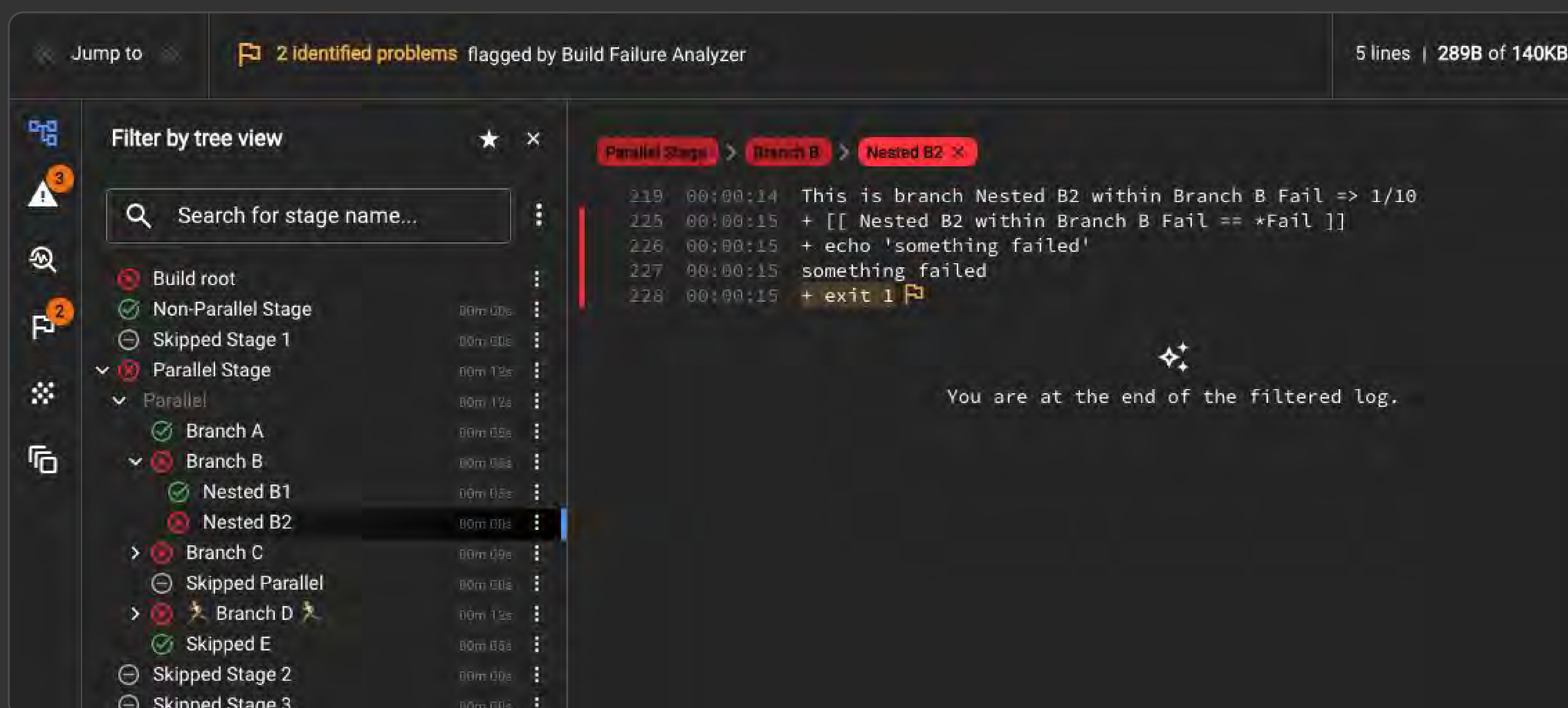
FINDING that starting point

Let's assume you see your build failed in GitHub - you click on details to figure out what went wrong and are magically whisked away to the CloudBees Pipeline Explorer page (see note above) . The Pipeline Explorer can navigate the stages of your pipelines in a tree view that lets you filter the logs by stages in nested groups. From this view - you can quickly see where the failures are by the red X indicators.

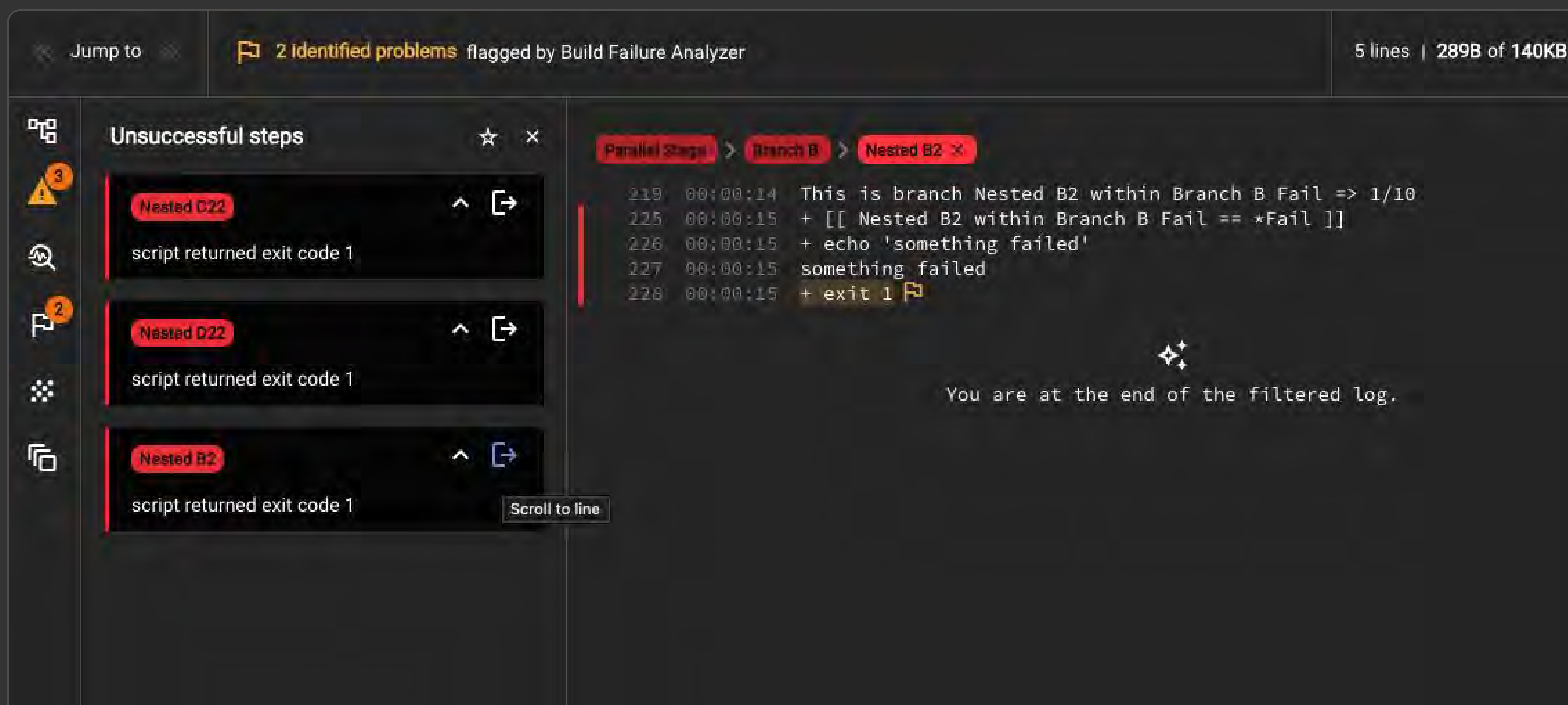


NOTE: Icon for TreeView  , clicking on this icon will expand or collapse this view.

- For this example - My pipeline has some nested parallelism and the failures are in my Branch B, C and D stages. You can drill down from my Parallel Stage and see how these errors trickled up. You can expand each Branch and see exactly where the failures are and select each failure to see this exact point in the log (failure in Branch B for example)



- Or, as an alternative, you can click on the Unsuccessful Steps icon and only see the errors in the panel:

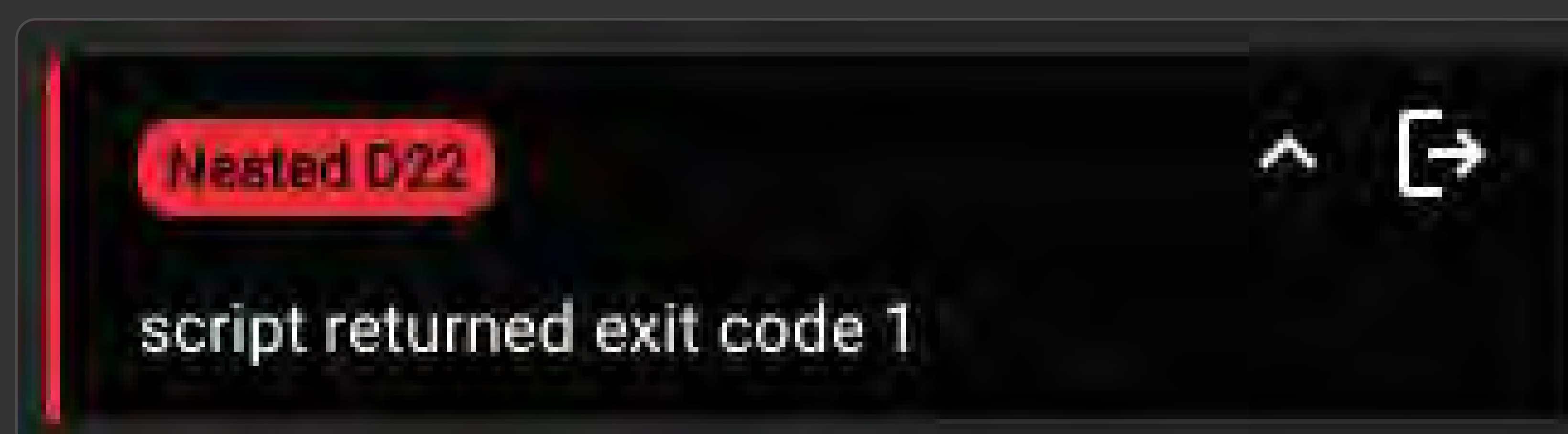


NOTE:  icon for Unsuccessful Steps, the orange circle indicates how many errors were found. Clicking this icon will expand or collapse this view.

Not my Pipeline, Not my Problem


As you can see - the errors in red are easy to pinpoint - but what if the error doesn't make sense or if it's in a pipeline owned by someone else? In this example - there are errors in the parallel branches B, C, and D, and you do not own the pipeline in Branch D, or perhaps - the errors you see do not make sense and you need some advice on how to fix them. There are several ways to share the data with a coworker quickly:

Click on the Investigating build failures with CloudBees Pipeline Explorer I will appear. I'm interested in what is going on in Branch D:

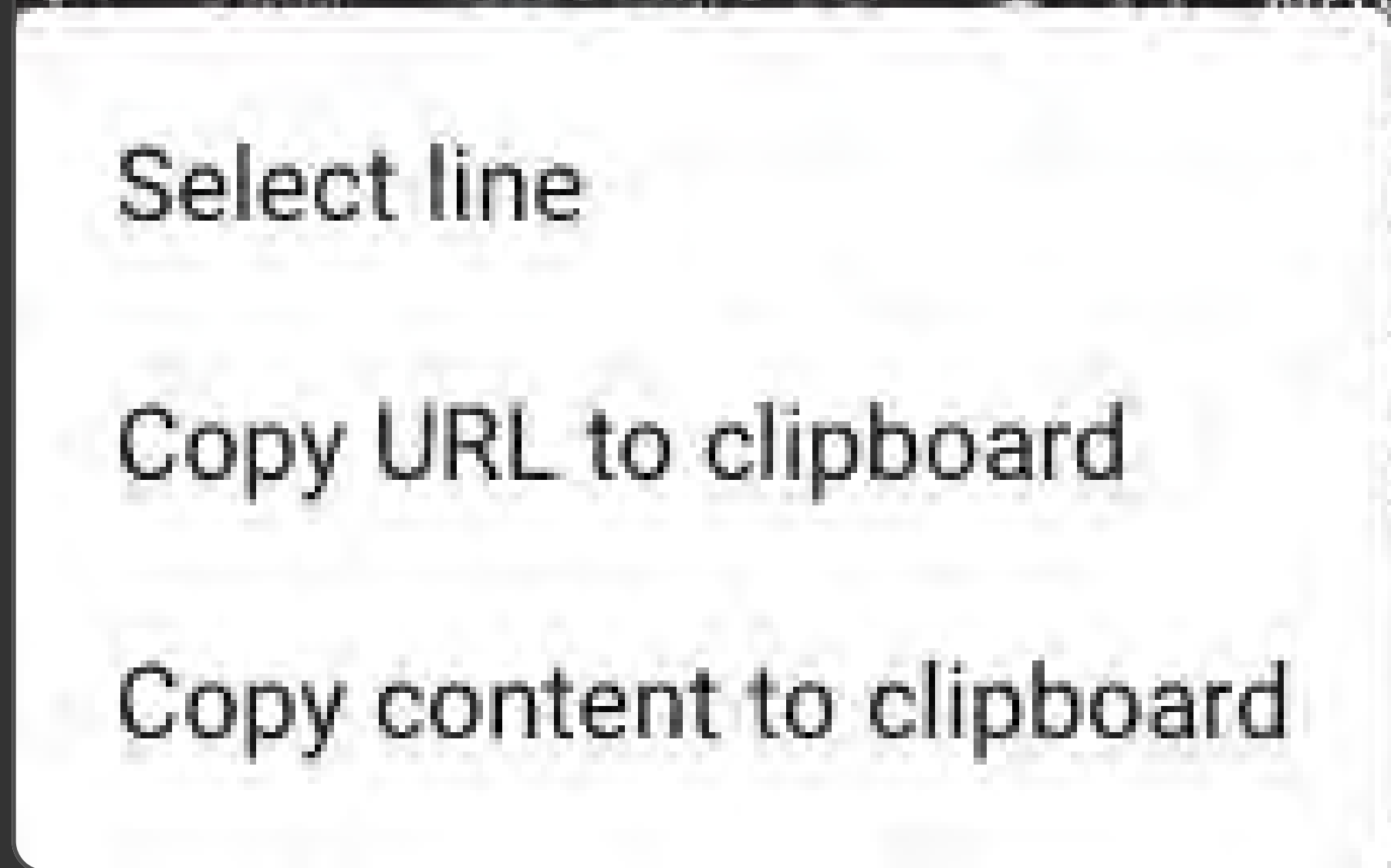


NOTE: Select the "scroll to the line" icon  and be brought directly to this error.

```
Parallel Stage > Branch D > Nested D2 > Nested D22  
299 00:00:19 This is branch Nested D22 within Branch D => 4/10  
300 00:00:19 + [[ Nested D22 within Branch D == *Fail ]]  
301 00:00:19 + '[' 1x = 1x ]'  
302 00:00:19 + echo 'something failed'  
303 00:00:19 something failed  
304 00:00:19 + exit 1
```

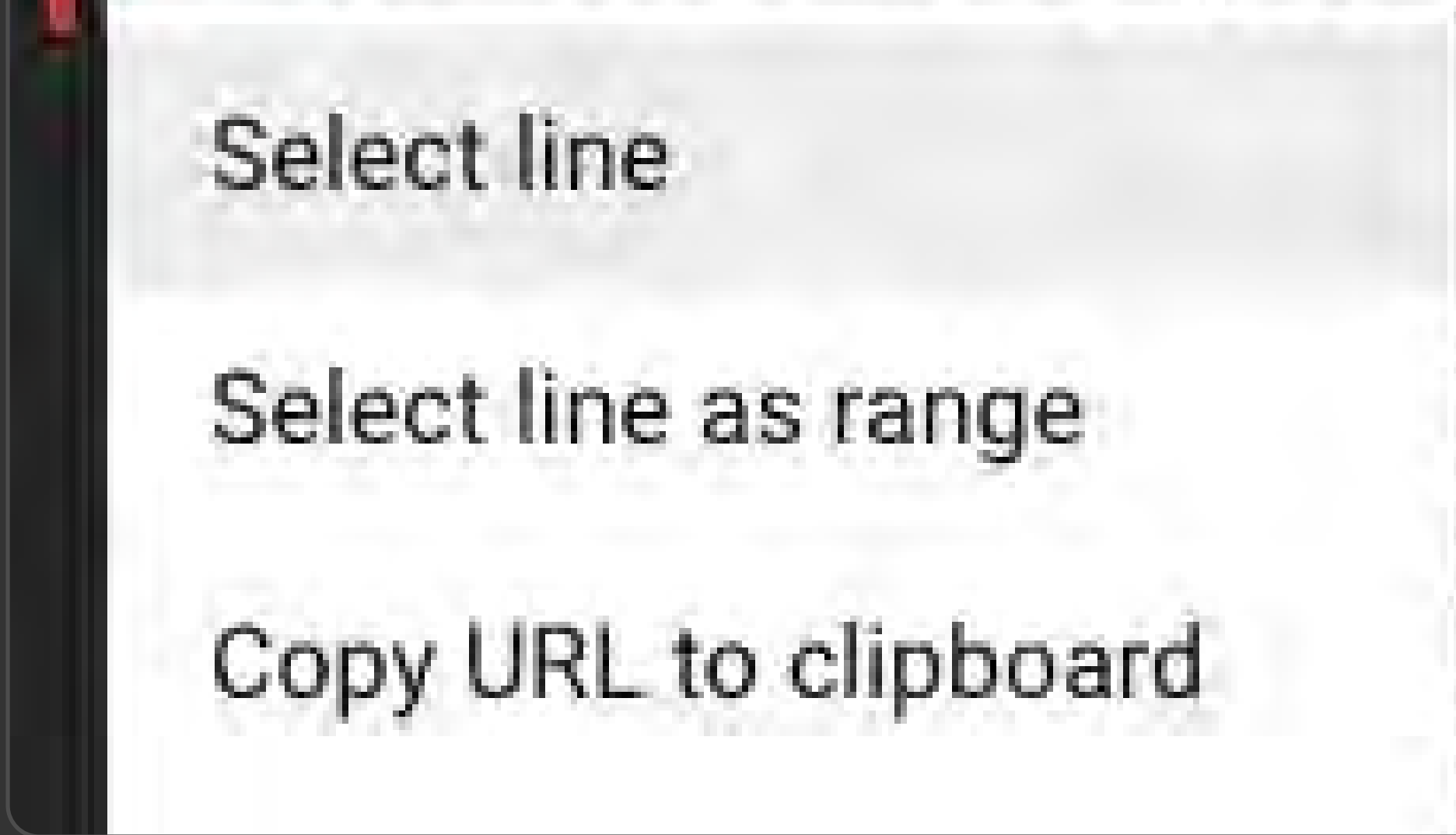
- There, you can select the line by hovering next to the line and clicking on the stacked dots  to copy the URL to the exact line of error:

```
Parallel Stage > Branch D > Nested D2 > Nested D22  
299 00:00:19 This is branch Nested D22 within Branch D => 4/10  
300 00:00:19 + [[ Nested D22 within Branch D == *Fail ]]  
301 00:00:19 + '[' 1x = 1x ]'  
302 00:00:19 + echo 'something failed'  
303 00:00:19 something failed  
304 00:00:19 + exit 1  
Nested D23" skipped due to earlier failure(s)
```



- You can also select a range of lines to share as well. Select the first line 299 through last line 304 and share this as a range of lines:

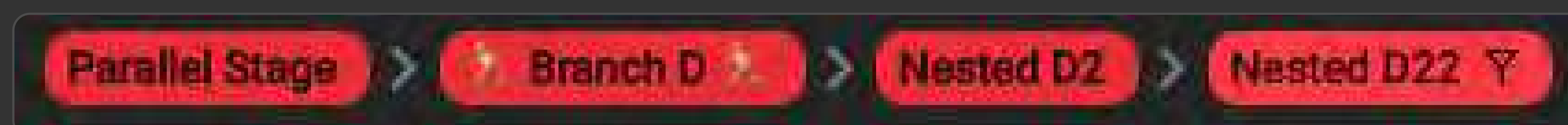
```
Parallel Stage > Branch D > Nested D2 > Nested D22  
299 00:00:19 This is branch Nested D22 within Branch D => 4/10  
300 00:00:19 + [[ Nested D22 within Branch D == *Fail ]]  
301 00:00:19 + '[' 1x = 1x ]'  
302 00:00:19 + echo 'something failed'  
303 00:00:19 something failed  
304 00:00:19 + exit 1  
Nested D23" skipped due to earlier failure(s)  
Nested D3" skipped due to earlier failure(s)  
Nested D3" skipped due to earlier failure(s)  
Nested D3" skipped due to earlier failure(s)  
Nested D3" skipped due to earlier failure(s)
```




- The same can be done from the tree view by clicking on the error and selecting the same stacked dots to capture the lines to share.
- Send this URL via slack, email - whatever works to get them to the right place quickly.

What if the person you need help from doesn't have access to Jenkins? Perhaps there is a Java expert in-house who could help with this specific Java error, or maybe you are looking for advice from the wizards on StackOverflow. How will you share this error without downloading the entire log and sharing this giant txt file for THEM to wade through?

- From either panel - TreeView or Unsuccessful Stage - you can filter out everything except the desired stage by hovering over it and selecting the filter icon next to it.

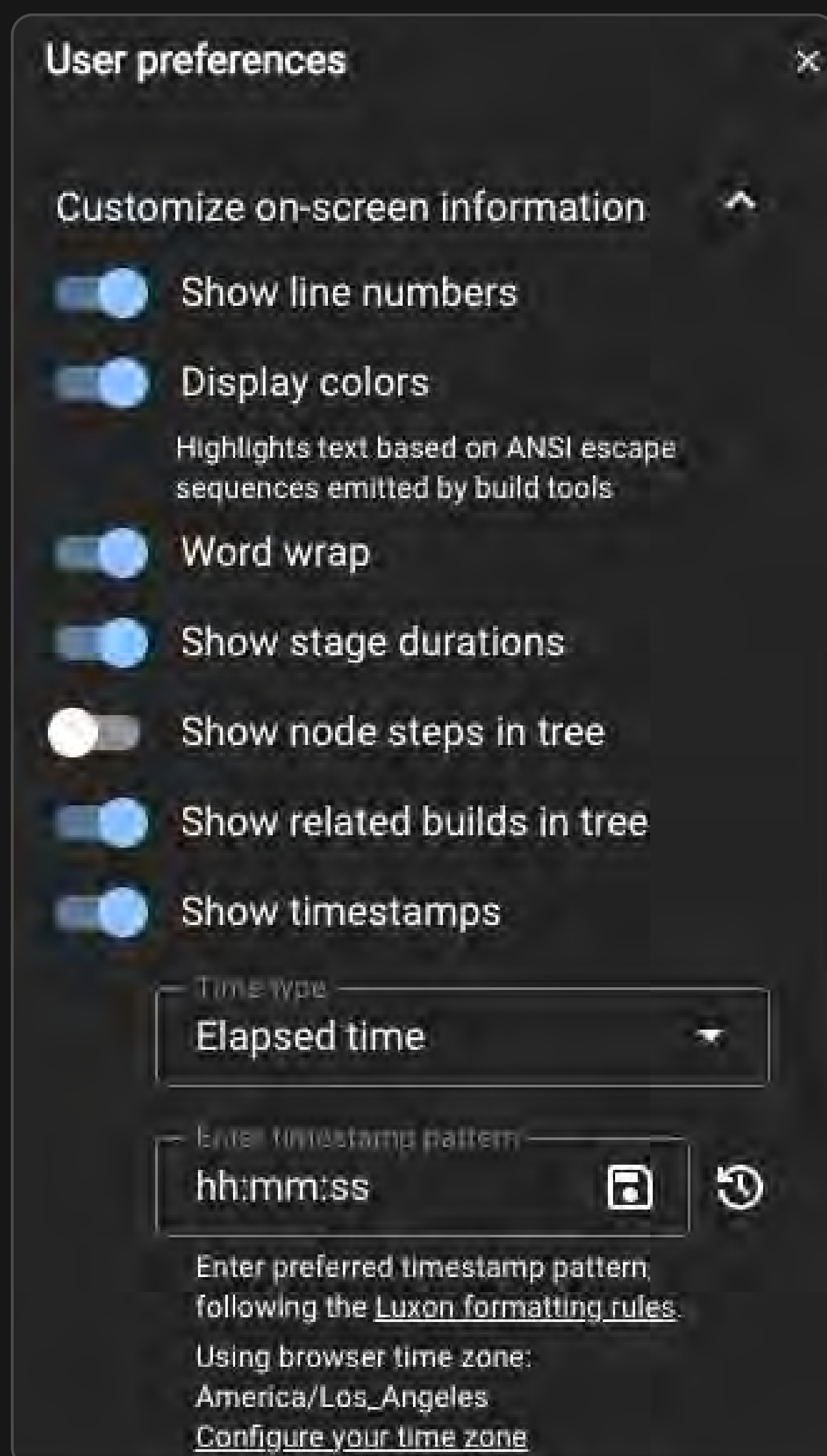


- Your view will be updated to display only this filtered stage, and you can now download  the smaller snippet to share/post. If your log is small to start with - maybe this is not an issue as you and screenshot or attach a small file- but if you have huge MB-sized log files, this gives you a focused log to share with those outside the Jenkins environment

Set your mode and preferences

The new pipeline explorer can easily get to the error quickly. Additionally, the log viewport can be customized with your preferences for line numbers, displaying ANSI colors from the log text, word wrapping, and custom time formats for timestamps, among other preferences. Let's run through a few of these customizations to get this log viewer to your visualization standards.

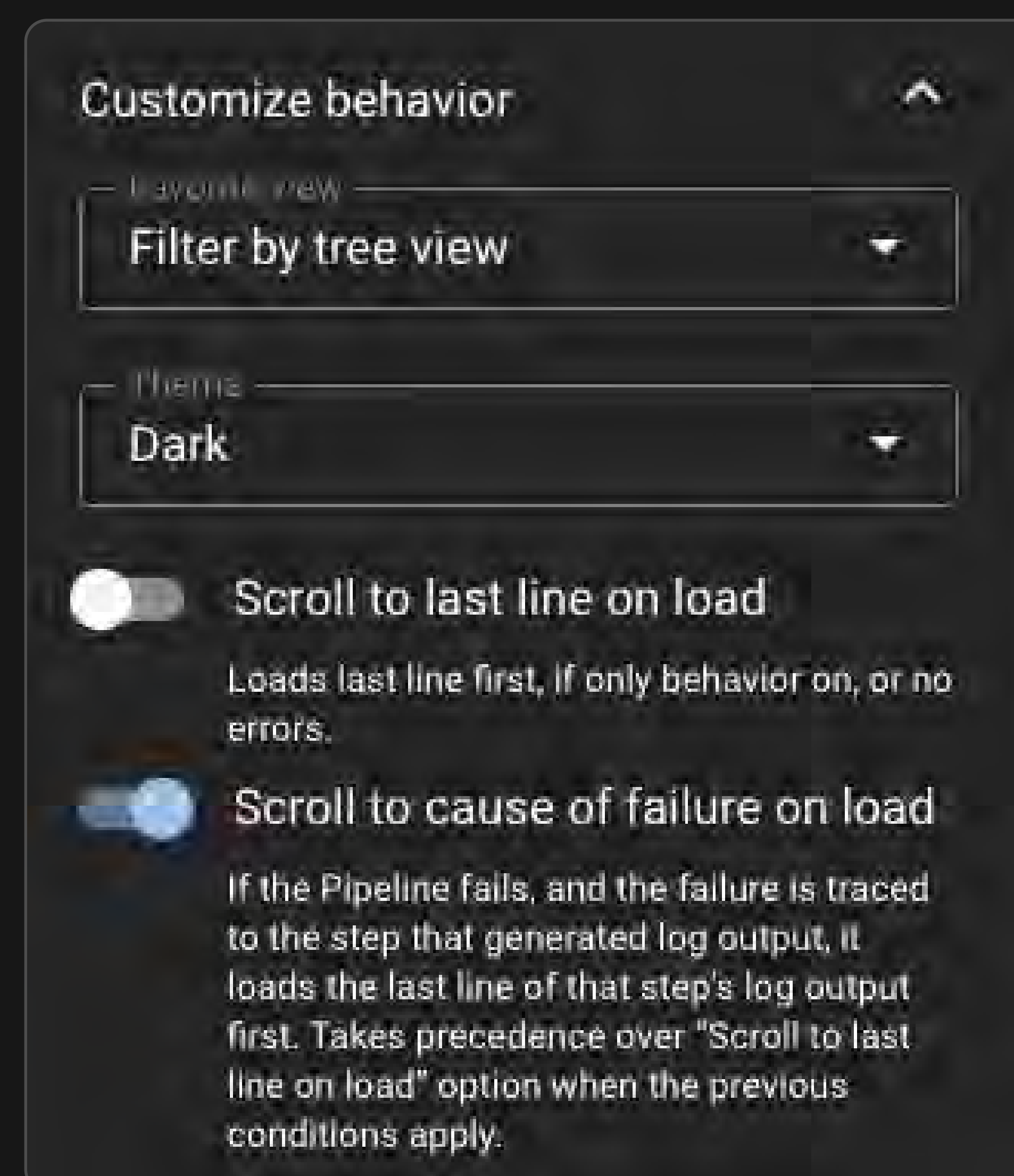
- Near the bottom of the left-hand panel - you see this familiar tool/settings icon:
- This is the User Preferences panel. Most as self-explanatory, but let me highlight a few:



- Show stage duration: Display the duration of each stage in the tree view
- Show node steps in tree: This option displays agents or nodes in the tree view. Some stages run on specific nodes/agents. So, if a specific stage is failing but on a specific node, it gives you a good indicator of where to check next.
- Show Timestamps: Choose between Absolute time (matching your system time) and Elapsed time(capturing your build duration time); set your timestamp pattern and timezone.

- Scroll further down in the User Preferences, and you will see the Customized behavior panel:

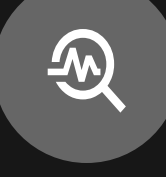

- Favorite View: Select your view here, or use the icons in the left panel to set your default view to tree view, Unsuccessful Steps view, etc.
- Theme: Light/Dark mode: I prefer dark mode because bugs like the light.
- Scroll to last line on load - no forever scrolling - get right to the end.
- Scroll to failure: This gets you to the root cause of the issue. If it is not turned on....lots of stuff to wade through. After it is turned on, BAM - right to the error. Regardless if you have parallel or sequential pipelines, the pipeline engine will know the root causes and get you right there.



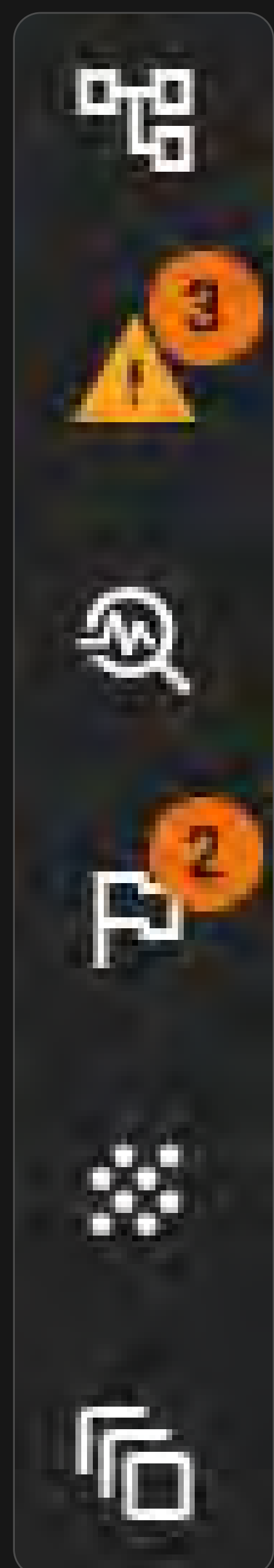
Other fun features: Artifacts, Related builds, Insights, and Search

You may have noticed some other icons in my screen capture examples. I walked you through the TreeView and the Unsuccessful Steps. What about the others?

The next two icons in this list are dependent on additional integrations. More on these later.

- Test Insights  requires the plugin J-Unit to be installed and enabled.
- Build Failure Analyzer  requires the plugin BFA to be installed and enabled.

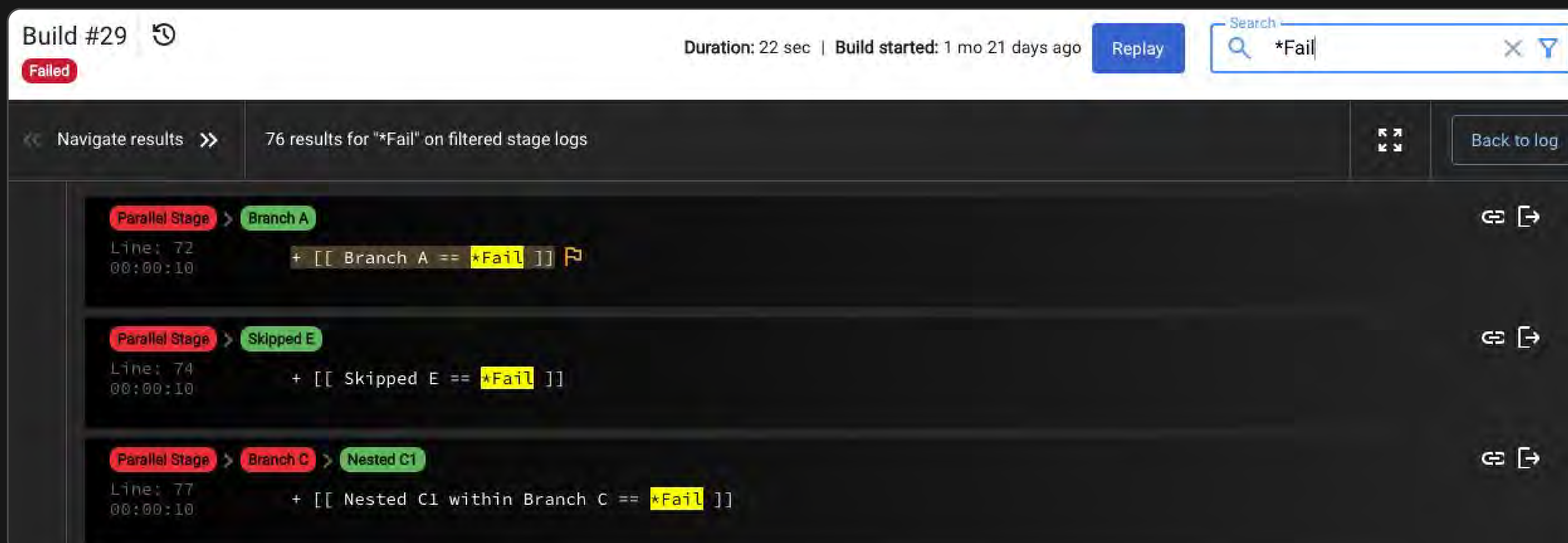
NOTE: These icons appear if you use additional J-Unit or BFA plugins. If you do not have these installed/enabled, you will not see these icons.



This brings us to the remaining icons of Build Artifacts  and Related builds  :

- Build artifacts: Directly access the build artifacts with a link to the build artifacts report. Less clicking around trying to find the artifact.
- Related Builds: Display the builds related to your current build. The tree view also displays the related builds in the stage where the build was triggered. The related builds include the builds triggered in the current build and the origin (parent) build that triggered the current build.

Finally - let's cover the Search. That's pretty clear what it does, so why mention it if I am brought directly to the error(s) I must address? Well, you and I both know - there's a lot of info within these logs themselves. Yes, I can be brought to the errors, but what if I want a summary of the errors with a specific fail code? I can search the whole log or a filtered set of the log based on my tree view selection - just to see my specific "*Fail" code:



In this example, my log file is small but if you have a large file - say - 700MB - you can use the filter to narrow down just the loglines you need - to share, send, or work with yourself.

NOTE: *Search is disabled by default and must be enabled by giving users the Run/SearchLog RBAC permission.*

Now that we have the basics, lets tackle some integrations with Build Failure Analysis (BFA) and JUnit and see what these look like in the explorer.

JUnit Integration with CloudBees Pipeline Explorer

JUnit: An Overview

Before we dive into the integration, let's clarify what JUnit is and why it's an indispensable tool for Java developers. JUnit is a simple yet powerful framework for writing automated, self-verifying tests in Java. Thanks to its support for test assertions, suite development, and immediate test reporting, it has become the most popular unit testing framework in the Java ecosystem.

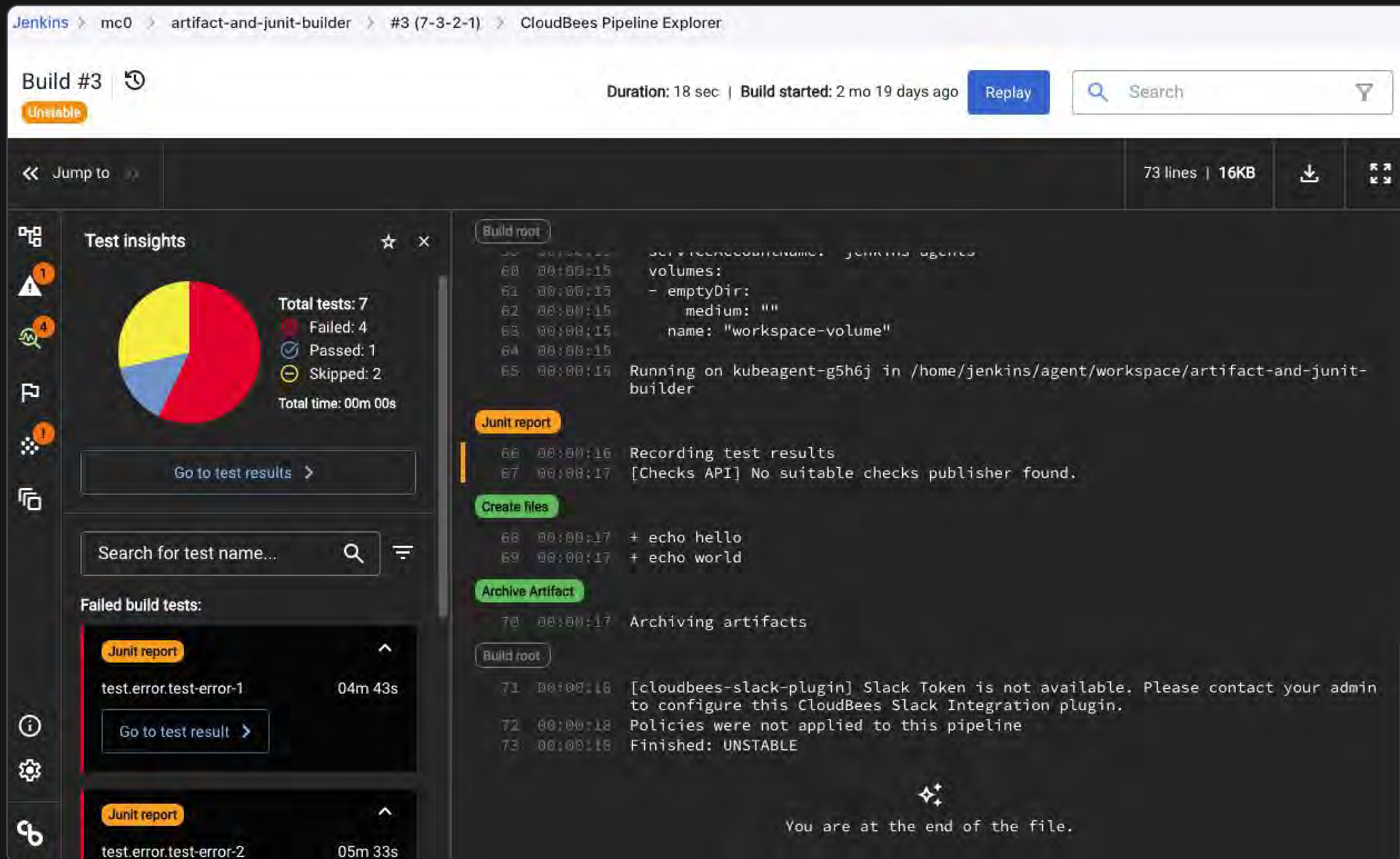
JUnit plays a significant role in regression testing, ensuring new code changes don't negatively affect previously written code. The framework provides useful information about test results, including historical trends, a user-friendly web interface for viewing test reports, tracking failures, and much more.

NOTE: *Read more about the Jenkins JUnit plugin*

Benefits of JUnit Integration into CloudBees Pipeline Explorer

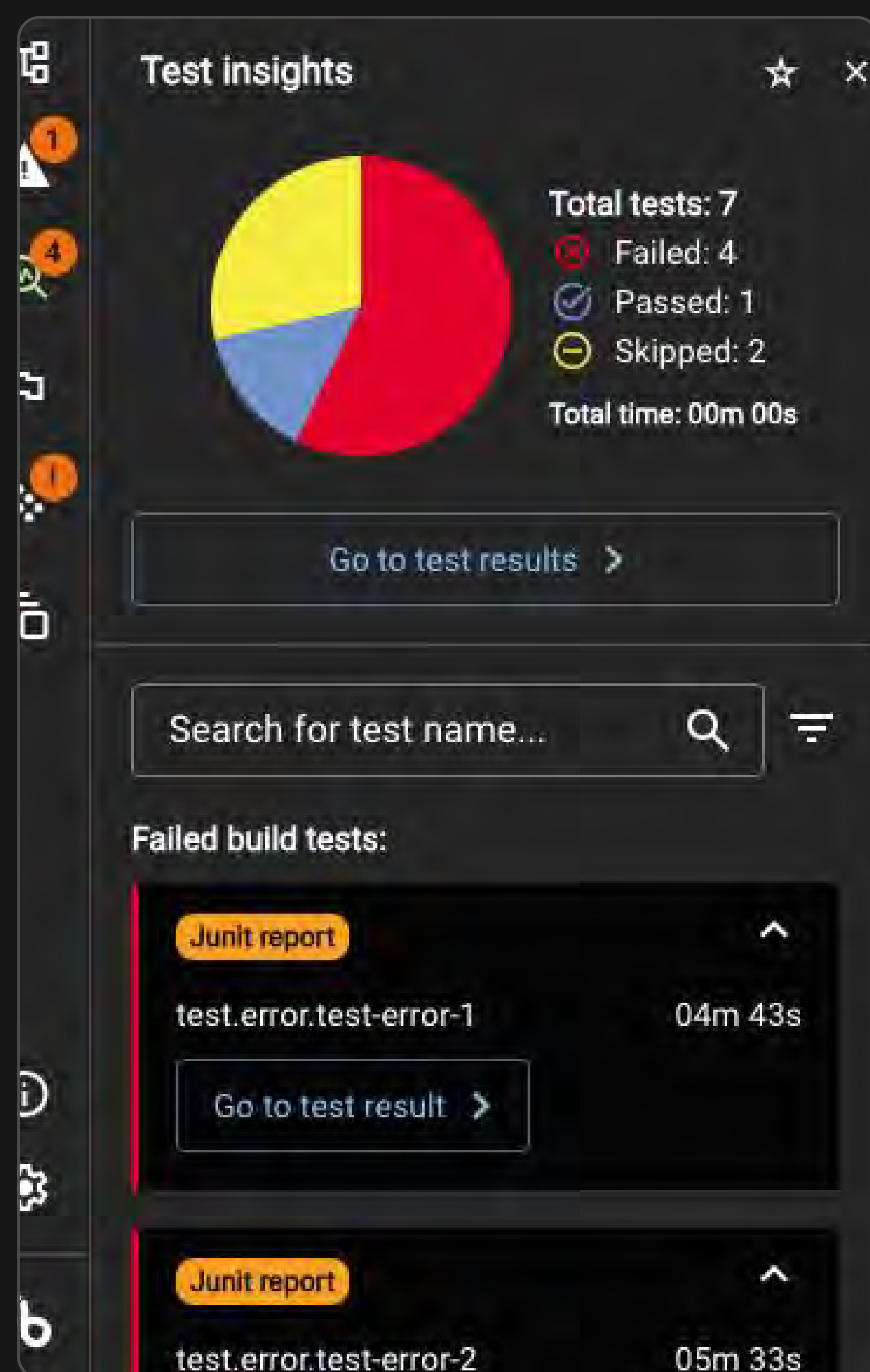
Now that we've reviewed JUnit's importance, let's discuss why integrating it into the CloudBees Pipeline Explorer (CPE) is helpful. Firstly, by integrating JUnit with CPE, you get a centralized place to visualize failed build tests.

This integration takes place under the test insight icon  in the Pipeline Explorer, leading you to the test insight panel shown below:



Although JUnit has its own dashboard, we've integrated it with Pipeline Explorer to highlight and consolidate what's in that dashboard, making your workflow more streamlined.

The test insights panel summarizes your test results and a list of test failures, including the stage each failure was in and a link to that test on the JUnit plugin page. You can search for tests and test class names, sort tests by name or duration, and display individual test results.



Moreover, the integration directly links to the failures on the JUnit page within Jenkins, offering more detailed insights into each failure. In essence, it provides you with an overview of statistics and visualizations directly in CPE by clicking the test insights icon.

NOTE: The JUnit plugin must be installed in your Jenkins environment to leverage these benefits. The test insights icon, which leads to the test insight panel, only appears if the JUnit plugin is installed.

Integrating JUnit with CloudBees Pipeline Explorer simplifies and enhances your testing process, providing a centralized, user-friendly interface for viewing and managing your test results. As developers, it's essential to have tools that make our jobs easier and increase our code's reliability and stability. This integration does just that, and we're excited to see how it improves your workflow!

Build Failure Analyzer Integration

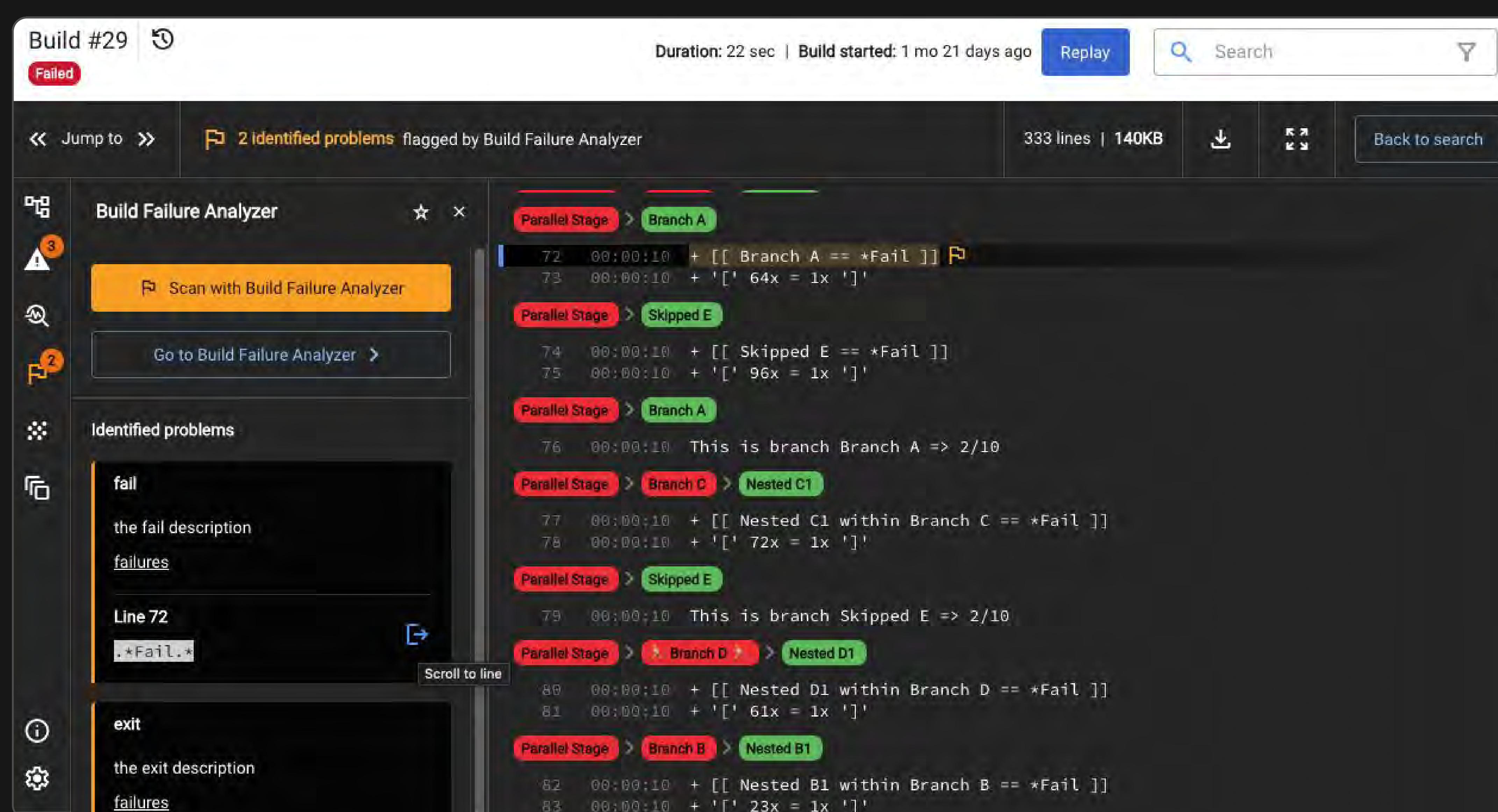
Jenkins Build Failure Analyzer (BFA) Plugin Overview

What is this BFA plugin you ask? It's a user-managed database of known error messages coupled with a scanner that checks these error messages against failed build logs. The Jenkins Build Failure Analyzer (BFA) plugin scans build logs and other files in the workspace for recognized patterns of known causes for build failures. It displays these on the build page to help you quickly pinpoint why it failed. The CloudBees Pipeline Explorer takes that data and provides visualization in the pipeline explorer.

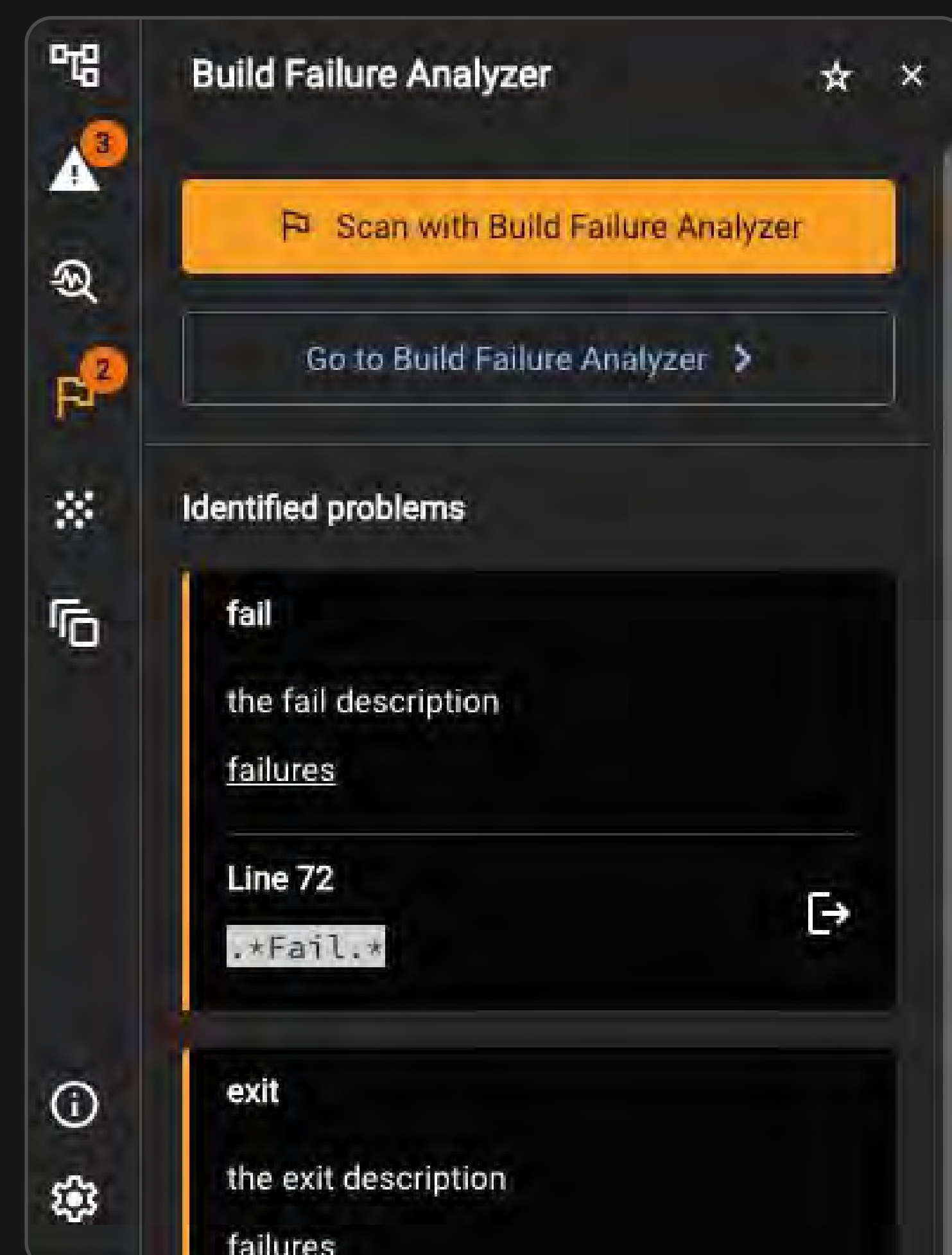
NOTE: *Read more about the Jenkins BFA plugin*


Benefits of BFA Integration into CloudBees Pipeline Explorer

Integrating BFA into CloudBees Pipeline Explorer means less clicking outside of the explorer. It consolidates your visualization in one place, making it easier to access and analyze information.



The Build Failure Analyzer (BFA) panel in the pipeline explorer displays results from the BFA plugin, presenting a list of identified problems. The annotated lines updated in the log view can be viewed. The identified problems display the name, description, categories configured in the BFA plugin, and related lines in the build. Want to access failed BFA builds? Just select the "Scan with Build Failure Analyzer" option. Here, you can scan either the non-scanned builds (the default) or all recurrent builds.



NOTE: *The Build Failure Analyzer plugin must be installed in your Jenkins environment to leverage these benefits. The Build Failure Analyzer icon  only appears if the plugin is installed. The orange circle indicates the number of identified problems*

The BFA plugin solves the painful process of sifting through logs to find error messages. You input a list of patterns to look for in the log and when a build fails, the plugin scans the log, looking for your input patterns. If it finds one of these patterns, it displays a message you've also input along with the pattern.

The beauty of this lies in its simplicity. If you have a long-standing build with specific error messages that are not easily identifiable or understandable, you can input a pattern like "error 2057". This could mean something like "switch the knob three times and rerun the build" or "contact this specific team to fix this specific problem."

You can even include contact info (email address, #slack channel, etc) in the database.

Conclusion

Developed by Jenkins enthusiasts for Jenkins users, the CloudBees Pipeline Explorer aims to enhance existing logviewer capabilities. Feedback highlighted that current tools fell short in efficiently diagnosing issues or errors in pipeline execution. Therefore, we crafted the CloudBees Pipeline Explorer to streamline troubleshooting, making it quicker, simpler, and centralized for an improved user experience.

NOTE: *To learn more - be sure to visit our technical documentation or visit us on CloudBeesTV to see the "Troubleshooting pipelines with CloudBees Pipeline Explorer" getting started video.*