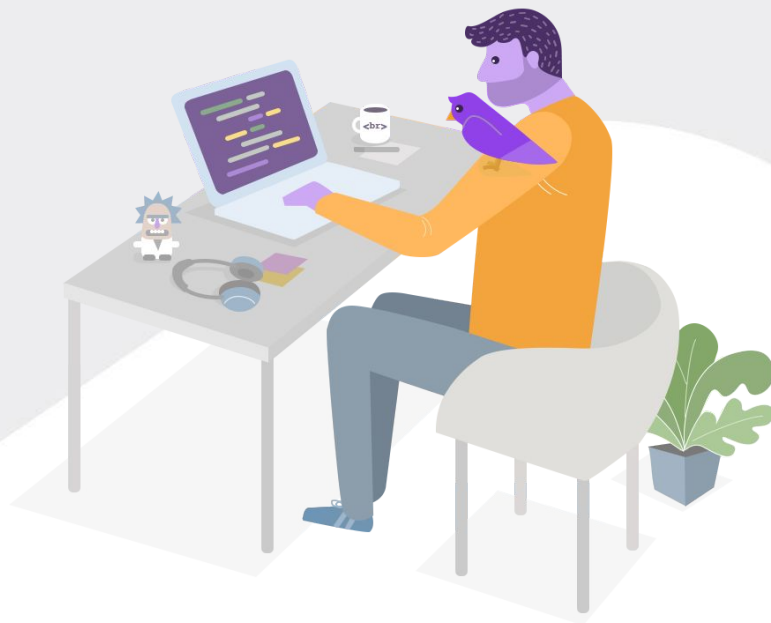




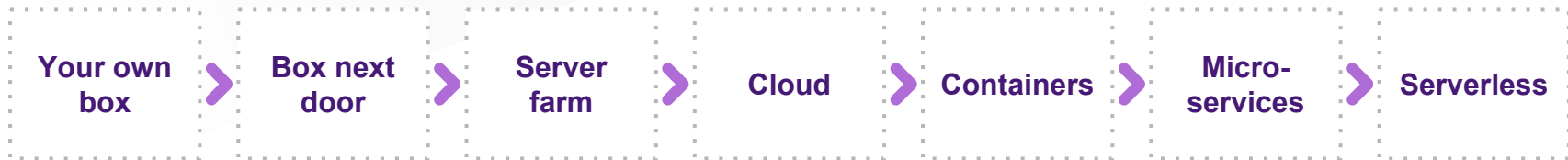
Live Code Observability

Or Weis, CEO

 [@orweis](https://twitter.com/orweis)



Servers: Wow, that escalated quickly!



The cost of flying blind



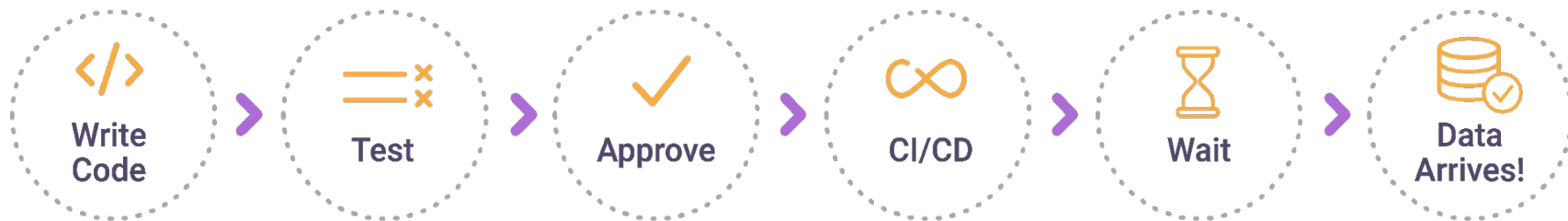
Average time spent on debugging, refactoring,
and modifying bad code:

17.3 hours per developer, every week (~%40)

* Stripe Research, September 2018

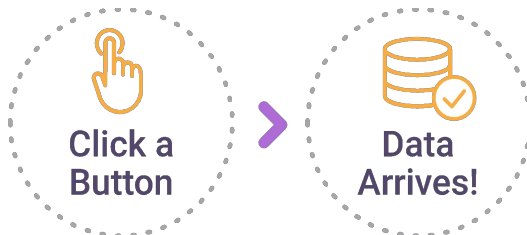


Why is debugging so time consuming?



- Lack of observability into live code data
- Misalignment between modern DevOps tools and old-school debugging methods

It doesn't have to be that way!



- Decouple live data visibility from the deployment process
- Cut debugging time by 80%! Free your dev team to focus on new features!
- Leverage the power of **non-breaking Breakpoints**

Customer Success



“ Rookout **accelerates our dev cycle** by helping us release features and fix bugs much more quickly. We now have a clear understanding of how the code behaves in our **microservice-oriented** architecture. ”

Nadav Sharir, VP Engineering



Customer Success



“Aldoc's app runs on a hospitals' local network and doesn't constantly send data back to our servers, so it's not always easy to trace a bug for a client on a different continent.

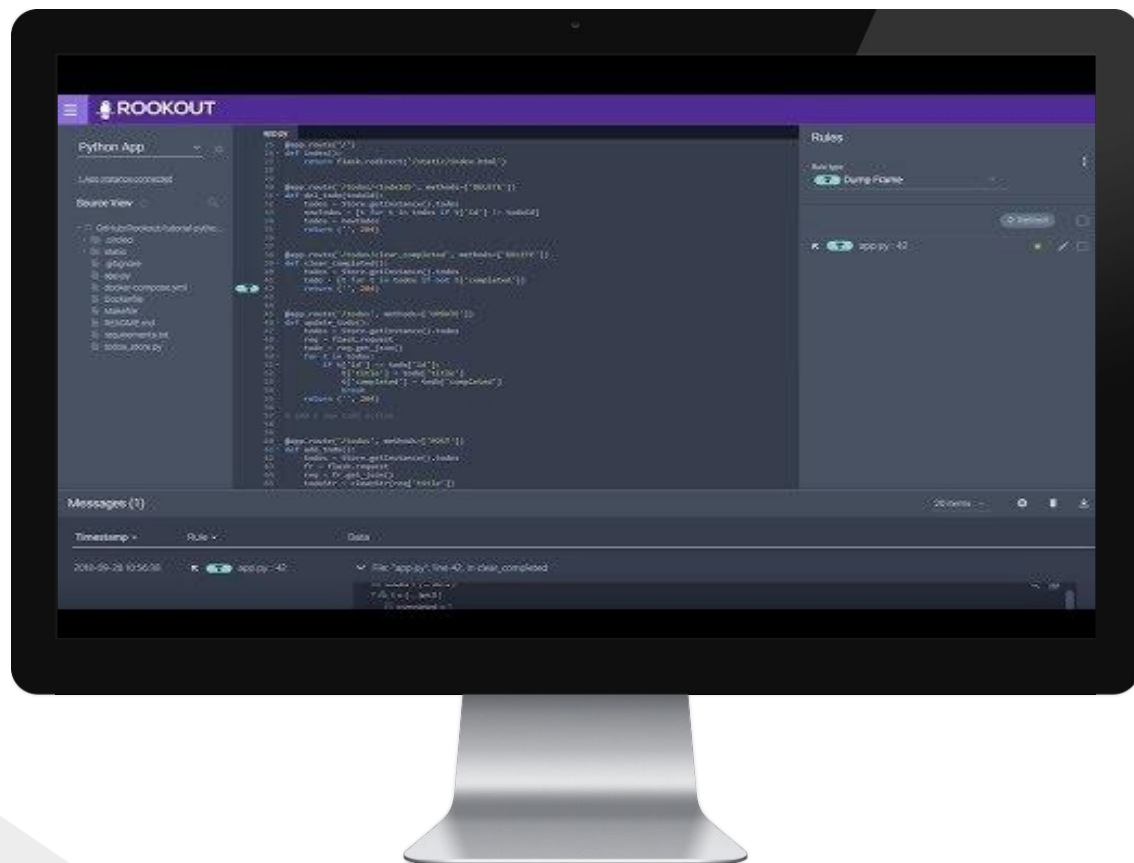
Rookout makes it possible to find bugs without spending hours on log-collection and trying to simulate the unique real-world environment of a particular hospital network

Guy Reiner, VP R&D

aidoc



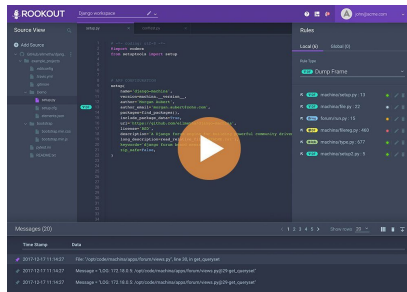
Rookout Demo



Videos and links



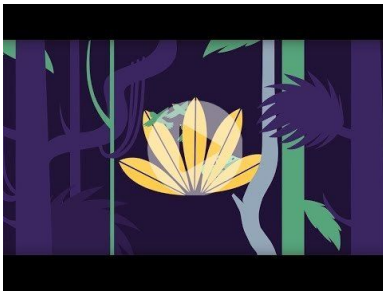
Basic demo



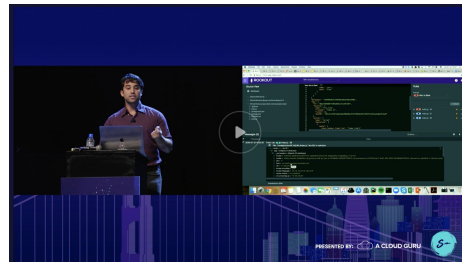
AWS Lambda Demo



"Your live code is a jungle"



Rookout at ServerlessConf



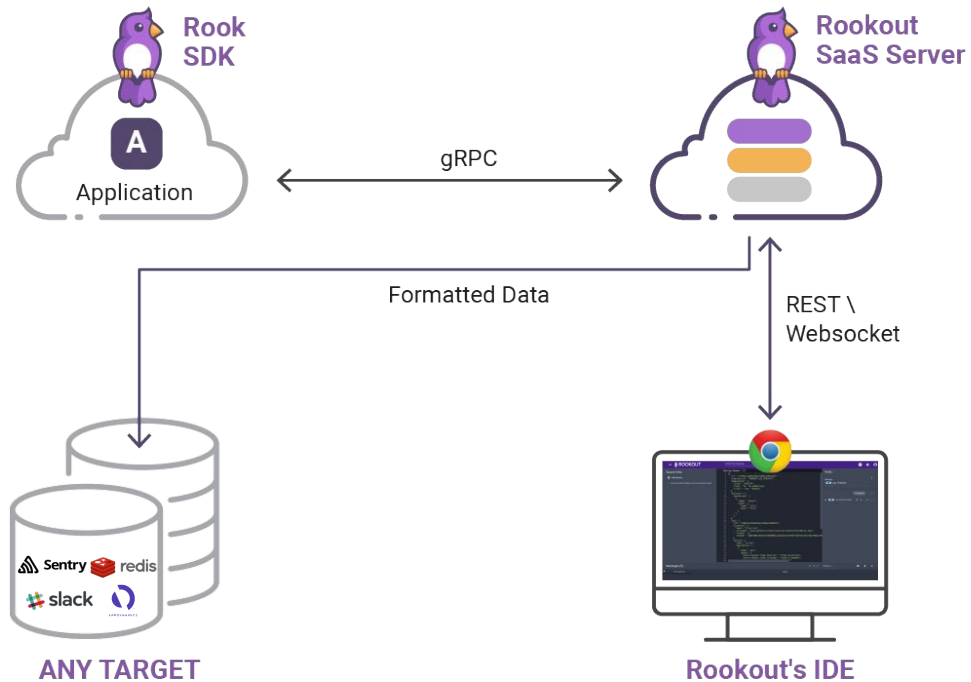
Select press coverage:

- [TechCrunch - Rookout fundraising](#)
- [TechCrunch - Rookout AWS Lambda Debugging](#)
- [ComputerWeekly - Production Debugging](#)
- [ComputerWeekly - Application Debugging \(ElectronJS\)](#)
- [Rookout CEO interview Sentry.io blog](#)

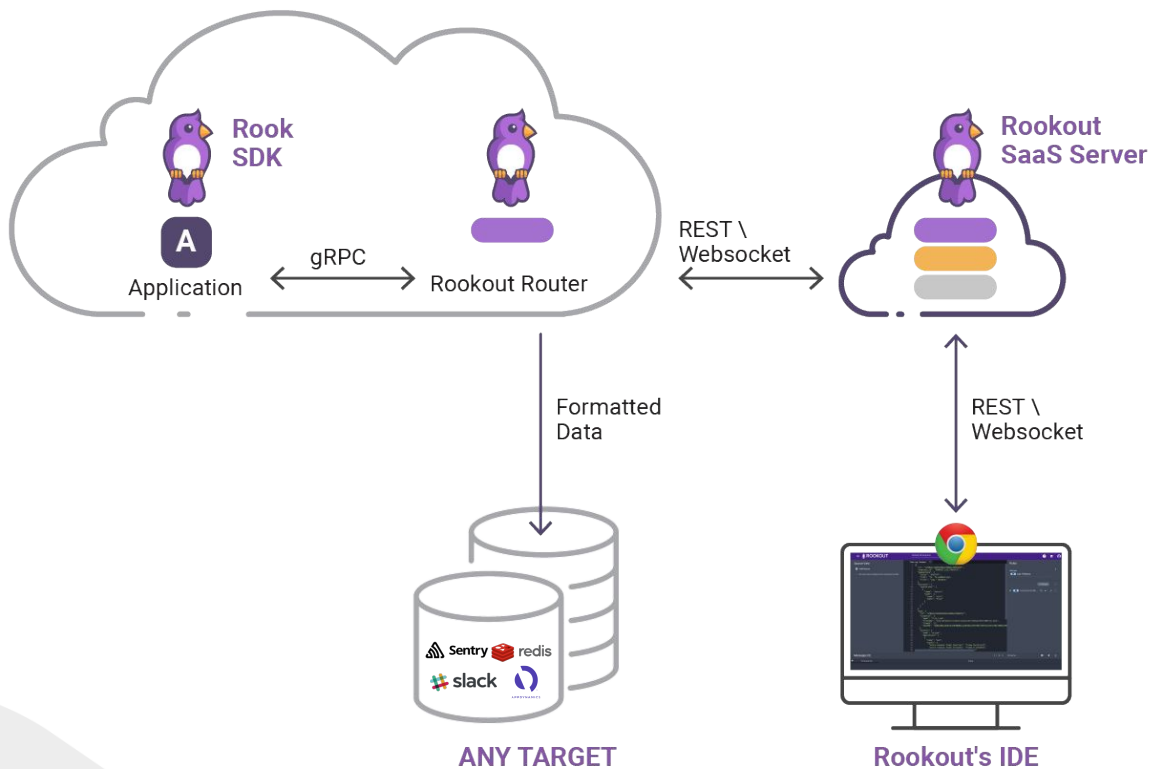
Links:

- [Rookout site](#)
- [Rookout Docs](#)
- [Brochure](#)

Architecture



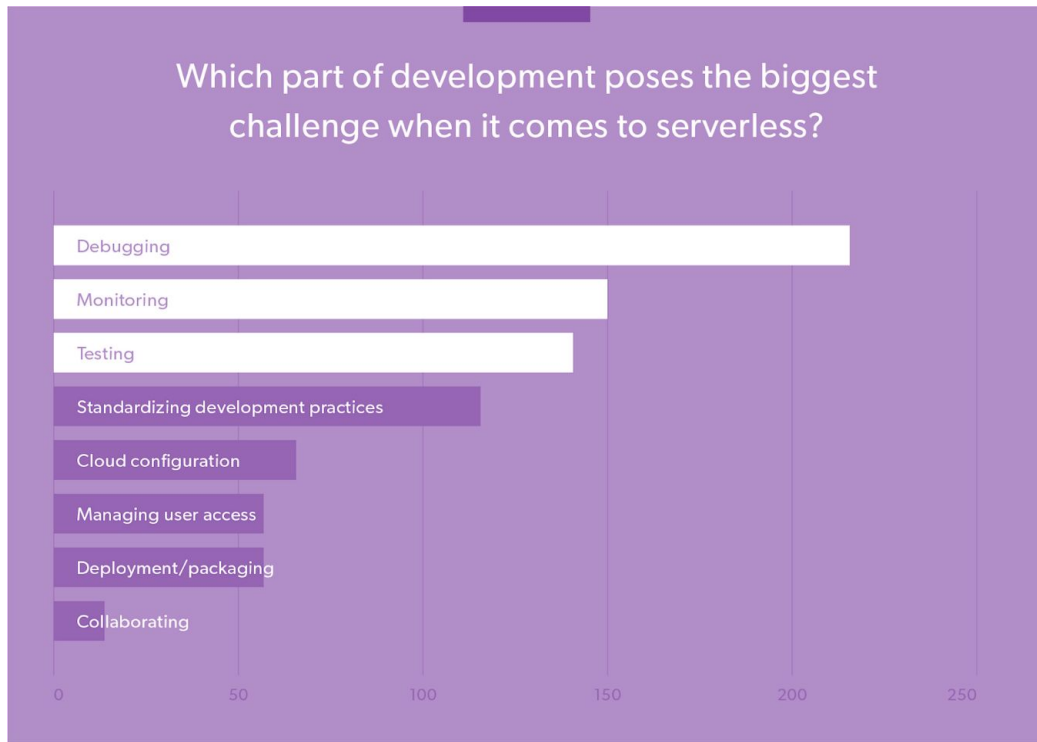
Data-On-Prem ('Router' Architecture)



The impact of slashing debugging time



Serverless.com survey: Debugging is the biggest challenge



Rookout benefits



- **Collect** data **on demand** from live code with just a few clicks
- Collect **in-depth** data - full trace, metrics, variables and more
- **No pre-instrumentation** required
- **Pipeline** data **anywhere** - APM, analytics, exception management, logs, alerts, storage...
- **Save hours** of developer work **every week**
- **Reduce** production debugging time **by 80%**

Free Trial: www.rookout.com/trial

Sample Use Cases



- Collect **ad-hoc data** for any **immediate need**
- Collect **long-term data** driven by product flows and needed alerts
 - Track features, exceptions, edge cases and more
- Solve **real-time issues in your production** environment
- Allow multiple devs to work on the same environments simultaneously
- **Improve support and R&D** communication, workflow, and handoffs
- **Debug everything** with a single pane of glass
 - **CI/CD flows, Serverless, Third-party** code

Security and compliance from the ground up



- Rookout is SOC2 (Type II) certified
- World-class security expertise and best-practices
- Full control of data flow, built in
 - Data-fencing
 - PII Redaction
- Data doesn't go through Rookout's servers (unless you want it)
- Your code NEVER goes through Rookout's servers
- Full on-prem option



Security and compliance from the ground up



- Rookout is SOC2 (Type II) certified
- World-class security expertise and best-practices
- Full control of data flow, built in
 - Data-fencing
 - PII Redaction
- Data doesn't go through Rookout's servers (unless you want it)
- Your code NEVER goes through Rookout's servers
- Full on-prem option



Collaboration Workspaces



- Online live collaboration
 - Users working in the same workspace **share a view** with the same rules and data
 - Better ticket handoffs
- Freedom to work
 - Users working in separate workspaces (even on the same infrastructure) **don't step on each others toes**
 - Save on duplicating environments

Workspaces

A workspace is an easy way to segment which apps and servers you wish to debug.

Type workspace name...

App Instances Filters (recommended)

Filter by platform, IP, host name, process name, or tags...

Suggested: `ip:100.103.10.124` `tags:staging` `processName:main.py`

5 App instances connected



Sources





No sources yet. Please add sources.

Cancel Apply













Rules approval process (Just like a Pull Request)



















  **ROOKOUT**

    johnsh@gmail.com

Pending Approval

Rule	Request	Workspace	Requested by	Request Time	Comment	
 machina/type.py : 677	 Create	Production	john@company.com	2017-12-17 11:14:27	Debugging issue 1317	 
 machina/type.py : 22	 Delete	Production	gil@company.com	2017-12-17 11:14:27	Hash mismatch	 
 machina/setup.py : 329	 Edit	Production	ben@company.com	2017-12-17 11:14:27	Handling production ticket with redirect issues	 


Active Rules (27) **Inactive Rules (1,457)**




Rule	Status	Workspace	Last modified by	Modification Time	Comment	
 machina/type.py : 201	 	Staging	ben@company.com	2017-12-17 11:14:27	Debugging issue 1317	
 machina/type.py : 154		Staging	ben@company.com	2017-12-16 11:11:21		
 machina/setup.py : 5		Staging	john@company.com	2017-12-16 10:14:17	Validating fix 1712 in staging	
 machina/type.py : 30		Django	jen@company.com	2017-12-14 13:12:00		
 machina/setup.py : 2		Production	john@company.com	2017-12-10 21:14:27		


< 1 2 3 4 5 >





Review a Rule Pull Request




 **ROOKOUT**

 johnsh@gmail.com

  machina/type.py : 22 In Production  

Time	Event	Event Activator	Comment		
2017-12-17 11:14:27	Created	john@company.com	Debugging issue 1317	View JSON	
294.	<pre>"id": "cb0f1c6702e348ee8c583acf052e0a13",</pre>				
295.	<pre>"location": {</pre>				
296.	<pre>"name": "file_line",</pre>				
297.	<pre>"filename": "src/server.js",</pre>				
298.	<pre>"lineno": 22,</pre>				
299.	<pre>"sha256": "964ee21b935887b7420f4e50d470539df3342c313188bd31b15916b1e9571821"</pre>				
300.	<pre>},</pre>				
301.	<pre>"action": {</pre>				
302.	<pre>"name": "script",</pre>				
303.	<pre>"operations": [</pre>				
304.	<pre>{</pre>				
2017-12-17 11:18:20	Approved	gil@company.com	This seems fine.		
2017-12-17 12:13:01	Edited	ben@company.com		View Changes	
2017-12-17 11:18:20	Changes approved	gil@company.com	This seems fine.		
2017-12-18 23:13:07	Last message received	Rookout	23 messages received for this rule		
2017-12-19 11:18:20	Rule status changed	Rookout	● Hash Mismatch		
2017-12-19 20:10:12	Deleted	ben@company.com			

Screenshots: “Just like a debugger, only live!”

The screenshot displays the Rookout web interface in a browser window. The interface is divided into several panels:

- Left Panel:** Shows the project structure for a Django application named 'machina', with a tree view of files and folders including 'apps', 'forum', 'migrations', and various Python files.
- Source View:** Displays the code for 'views.py' in the 'forum' app. A green cursor is positioned at line 29, which is the start of the 'get_queryset' method. A small green icon with a white 'i' is visible next to the line number.
- Right Panel:** Titled 'Rules', it shows a 'Rule type' dropdown set to 'Dump Frame'. Below this, there is a 'Refresh' button and a list of rules, with one rule selected: 'machina/apps/forum/views.py : 29'.
- Messages (7):** A table with columns for 'Timestamp', 'Rule', and 'Data'. The first message is from '2018-10-19 16:42:44' and is associated with the rule 'views.py : 29'. The data is expanded to show the local variables and context for the 'get_queryset' method, including 'self', 'args', 'kwargs', 'request', 'cached_user', 'messages', 'post_parse_error', 'read_started', 'stream', 'content_params', 'content_type', and 'COOKIES'.

Screenshots: "Send data anywhere with a click!"



The screenshot illustrates the Rookout interface and its integration with Slack. On the left, the Rookout web application shows a Django project structure with a file explorer on the left, a code editor in the center, and a 'Rules' panel on the right. The code editor displays the `views.py` file, which includes imports for Django and custom models, and a `get_queryset` method. The 'Rules' panel shows a rule named 'Alert to Slack' that triggers an alert when a file is changed in the `forum/views.py` file. On the right, a Slack channel named '#monitoring' shows a conversation. A Rookout alert is triggered by a file change in the `forum/views.py` file, and the alert is sent to the Slack channel. The alert message includes the file path and the change details.

Side by Side Rookout and Slack