

A Day in the Life: Mastering the RAD Support Workflow

An empowering guide for RAD
Platform Support Professionals.

The Ticket Arrives

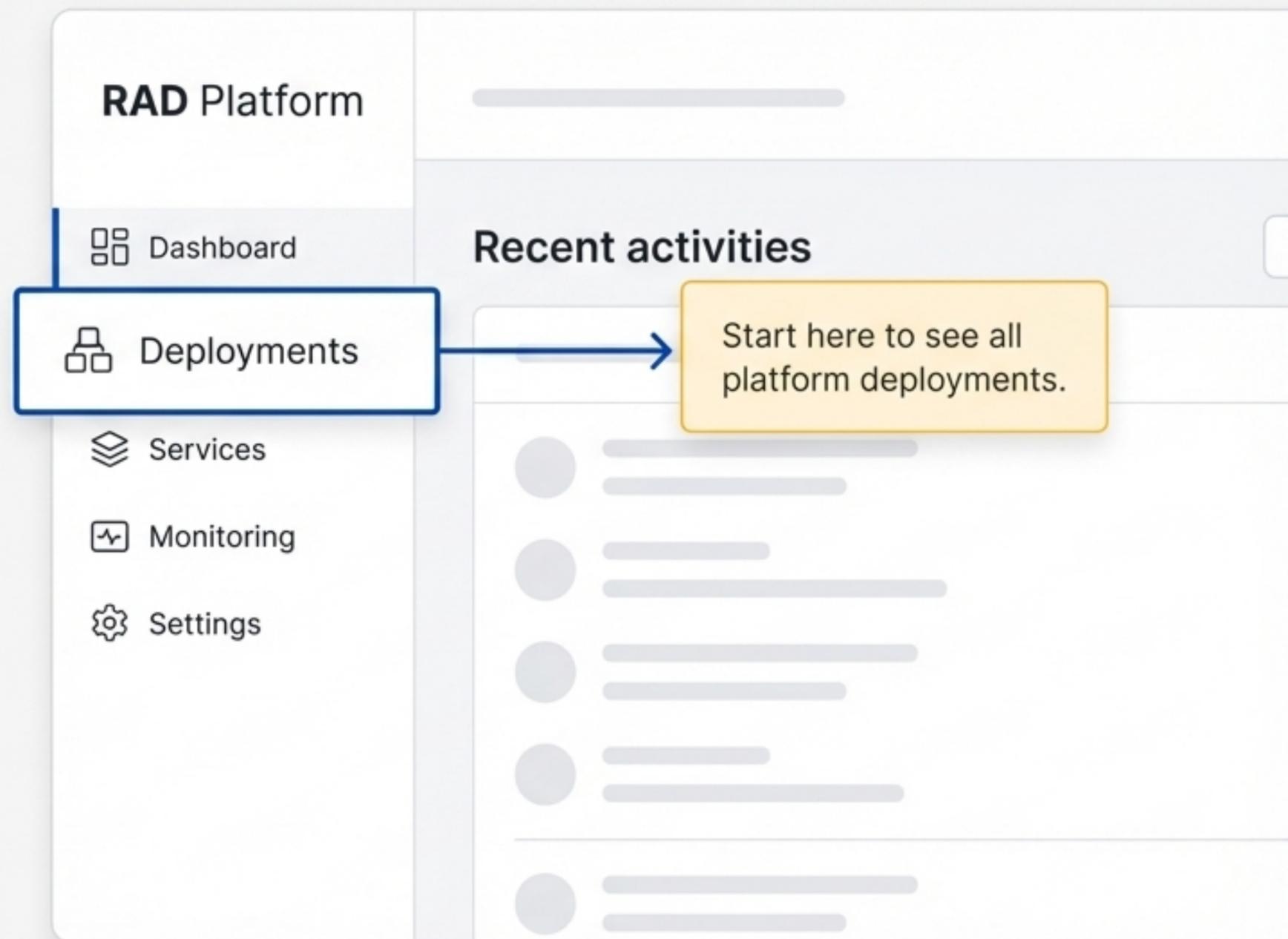
9:15 AM. A new ticket hits the queue. A user's deployment has failed, and they need answers. Your investigation begins now.

Subject	Deployment Failure
User	user@company.com
Deployment ID	d-xyz123
Status	FAILED

"My build is failing, please help! I don't know what went wrong."

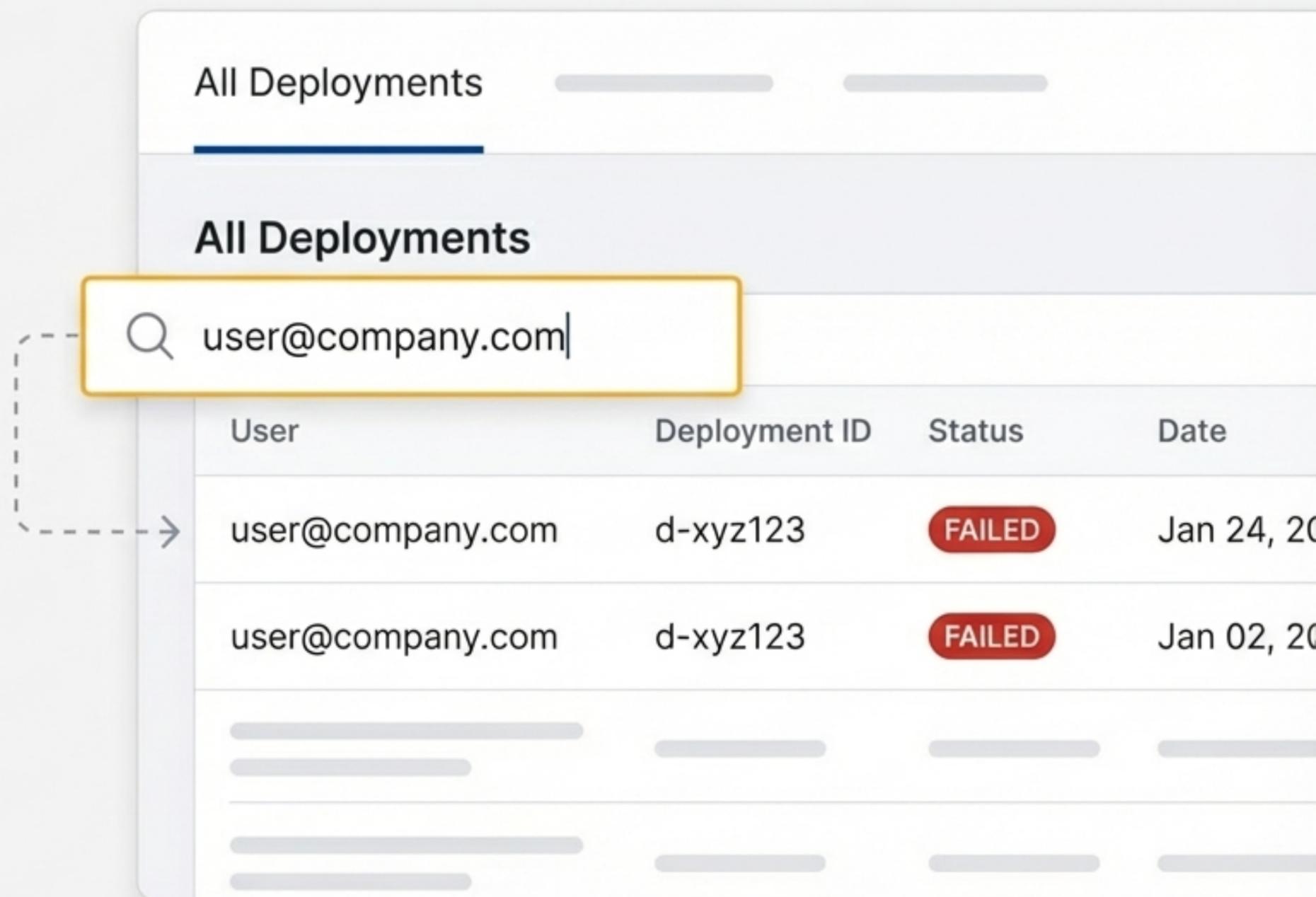
Step 1: Locate the User's Deployment

Your first move is to find the user's activity on the platform. The Deployments section provides a comprehensive view of all user activity.



Zeroing In: Search and Filter

Use the user's **email address** or the **Deployment ID** from the ticket to instantly filter the list and find the exact record you need.

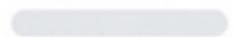
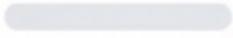


The screenshot shows a web interface for managing deployments. At the top, there is a tab labeled "All Deployments". Below the tab is a search bar containing the text "user@company.com". A dashed line points from the search bar to the first two rows of a table below. The table has four columns: "User", "Deployment ID", "Status", and "Date". The first two rows show a failed deployment for the user "user@company.com" with ID "d-xyz123". The first row is dated "Jan 24, 20" and the second is dated "Jan 02, 20". The remaining rows in the table are blurred.

User	Deployment ID	Status	Date
user@company.com	d-xyz123	FAILED	Jan 24, 20
user@company.com	d-xyz123	FAILED	Jan 02, 20

Step 2: Open the Deployment Details

Failed deployments are clearly marked. Click on the Deployment ID to open the detailed view, which contains the build logs and all other relevant information.

User	Deployment ID	Status	Date
	d-xyz123	FAILURE	Dec 20
	d-xyz123		Aug 20
	d-xyz124		Sep 20

Click to open the case file.

Cracking the Case: Analyzing the Build Logs

The truth is in the logs. Scroll down to the **Build Logs** section to find the specific error message that caused the failure.

```
< Deployment Details
-----
Upnaic details
-----
Build Logs
-----
12:13:59 AMM Building process...
12:13:59 AMM System+Invitation: Deployment error-arning failur...
12:13:59 AMM success:: Stat aeacted.
12:13:59 AMM System+Invitation: conarous.tsos/espoy/trwøslast.ap/oun.py'
12:13:59 AMM System+Invitation: wzia-rookageaies/deployment8icals...
12:13:59 AMM success:: Poot completed.
12:13:59 AMM success:: Staring initialize...
12:13:59 AMM success:: Staring agent...
12:13:59 PMM success.tring successfully.
12:13:59 AMM System+Invitation: cenarous.tsas/espoy/trnaslast.ap/con.py'
12:13:59 PMM System+Invitation: wbia-rookageaies/deploymentSisals...
12:13:56 AMM Error: Dseasöt Run error message: '/Users/appidata/pasowthyV0LPreparasveet/ientfiss/ierros?':
12:13:59 PMM success:: Poot completed.
12:13:59 AMM success:: Staring initialize...
12:13:59 AMM success:: Staring agent...
-----
ScrollInpy
```

The error message is located here.

The Usual Suspects: Common Error Patterns

Most failures fall into one of three categories. Learning to spot these patterns is the key to a fast resolution.



Invalid Input

The user provided a variable value that Terraform rejected.



Permissions

The service account lacked the necessary permission to create a cloud resource.

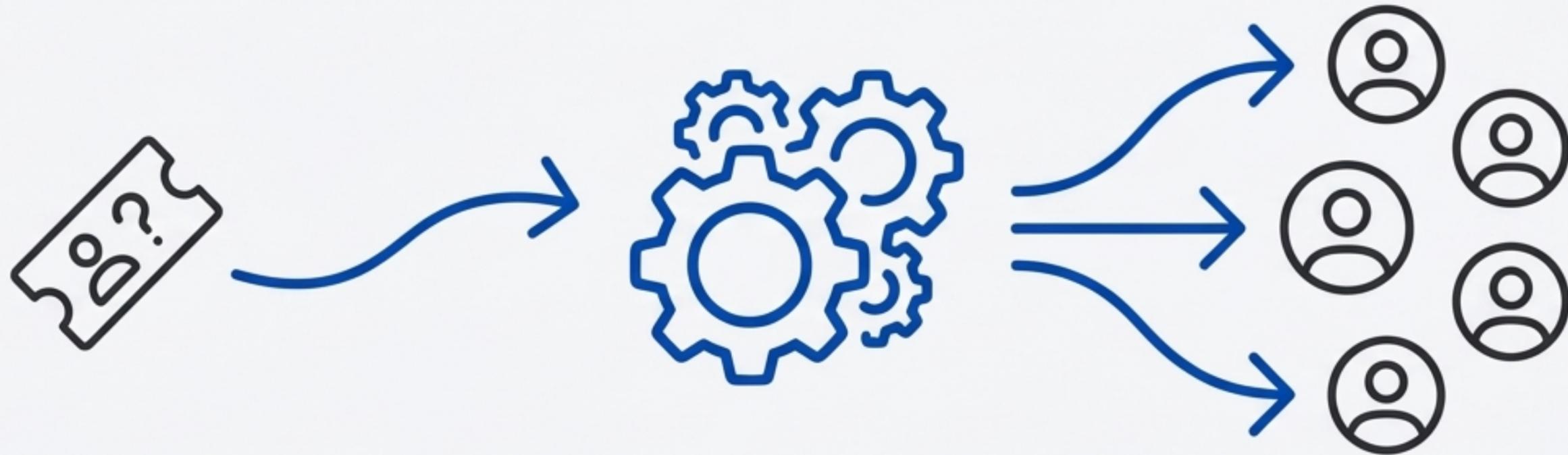


Quotas

The target cloud project has reached a resource limit (e.g., max number of VMs).

Level Up: From Reactive Fix to Proactive Solution

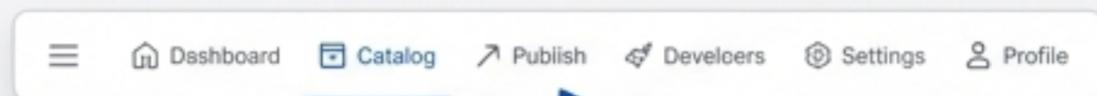
Solving one ticket is good. Preventing the next ten is better. As a Support user, you have the same module management permissions as a Partner. You can publish updates and roll out fixes for everyone.



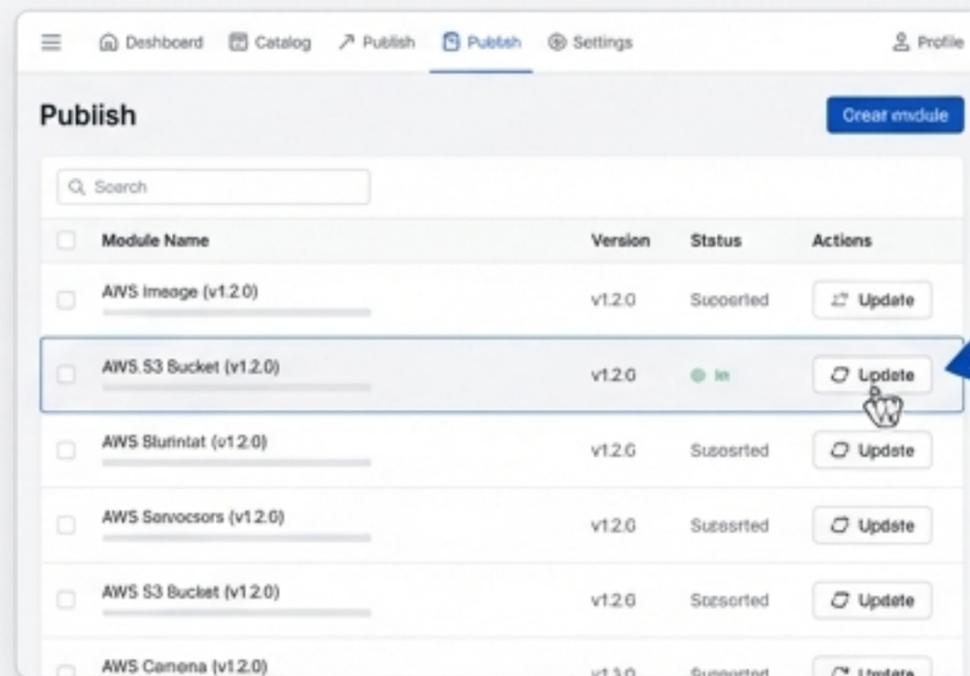
The Pro's Toolkit: Managing Modules

When a fix requires a code change, you can update the module directly in the catalog. (Note: Ensure your GitHub settings are configured in your **Profile** first.)

1



2



Click to refresh the module definition and roll out a fix.

Your Secret Weapon: The AI Debugging Assistant



Jules

For the toughest cases, you have an AI partner. Jules can analyze module code, explain complex errors, and suggest fixes, dramatically speeding up your debugging process.

Putting Jules to Work

On the **Publish** page, find the problematic module and click the Sparkles icon to start a session. (Note: Requires a **Jules API Key** configured in your **Profile**.)

The screenshot displays the 'Publish' page of a web application. The navigation bar includes 'Dashboard', 'Catalog', 'Publish', 'Publish', and 'Settings'. The main content area features a search bar and a table of modules. The table has columns for 'Module Name', 'Version', 'Status', and 'Actions'. The 'AWS S3 Bucket (v1.2.0)' row is highlighted, and a yellow circle with a sparkles icon is overlaid on the 'Update' button for that row. A chat window is open on the right, showing a message from Alex asking Jules for a fix for a permissions error.

Module Name	Version	Status	Actions
AWS Imagee (v1.2.0)	v1.2.0	Succeeded	Update
AWS S3 Bucket (v1.2.0)	v1.2.0	Failed	Update
AWS Bluntnat (v1.2.0)	v1.2.0	In Progress	Update
AWS Sarrocsors (v1.2.0)	v1.2.0	Supported	Update
AWS S3 Bucket (v1.2.0)	v1.2.0	Supported	Update
AWS Camona (v1.2.0)	v1.2.0	Supported	Update
AWS Camona (v1.2.0)	v1.3.0	Supported	Update

Chat window content:

Alex: Jules, can you suggest a fix for a permissions error related to creating this resource?

The Complete Support Workflow: From Diagnosis to Resolution

You have the tools to not only solve any user's issue but to proactively improve the platform for everyone. You are the expert.

