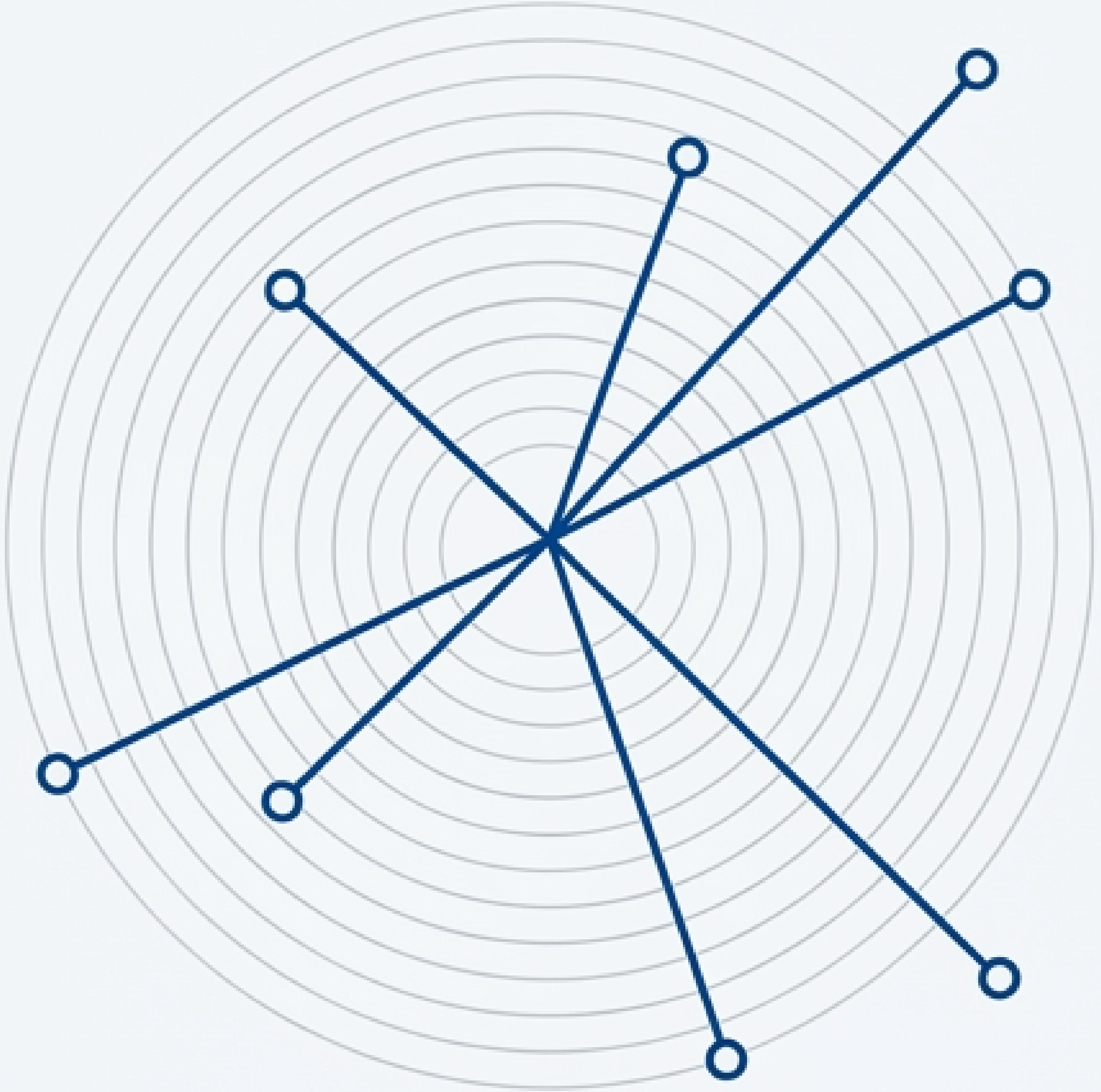


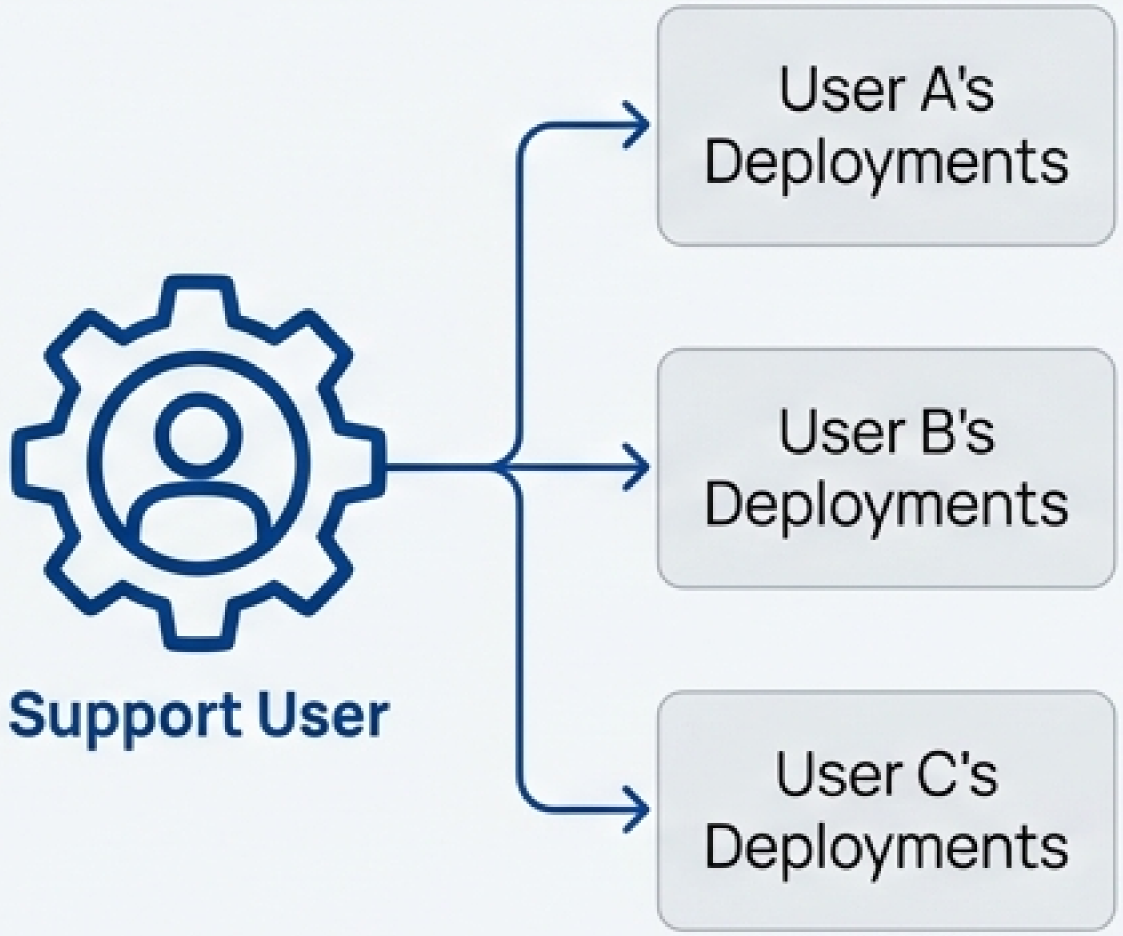
Mastering Support on the RAD Platform

Your Playbook for Troubleshooting and Resolution

Your Mission: Visibility is Your Superpower

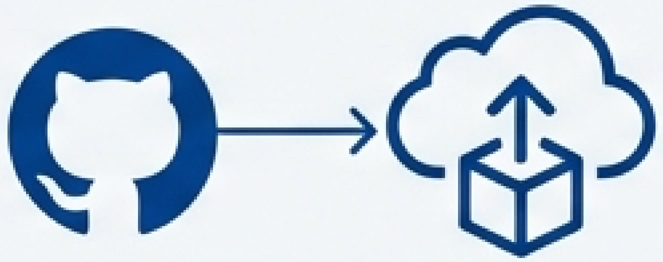
As a Support Specialist, you have enhanced privileges designed for a single purpose: to resolve user issues effectively. You see what others can't.

X-Ray Vision



You can view **All Deployments** across the entire platform. This is your primary tool for troubleshooting any user's issue, as you see their complete deployment history by default.

Publisher Access



You have the full capabilities of a Partner. This means you can connect a Git repo, publish modules, and update them, allowing you to test fixes directly.

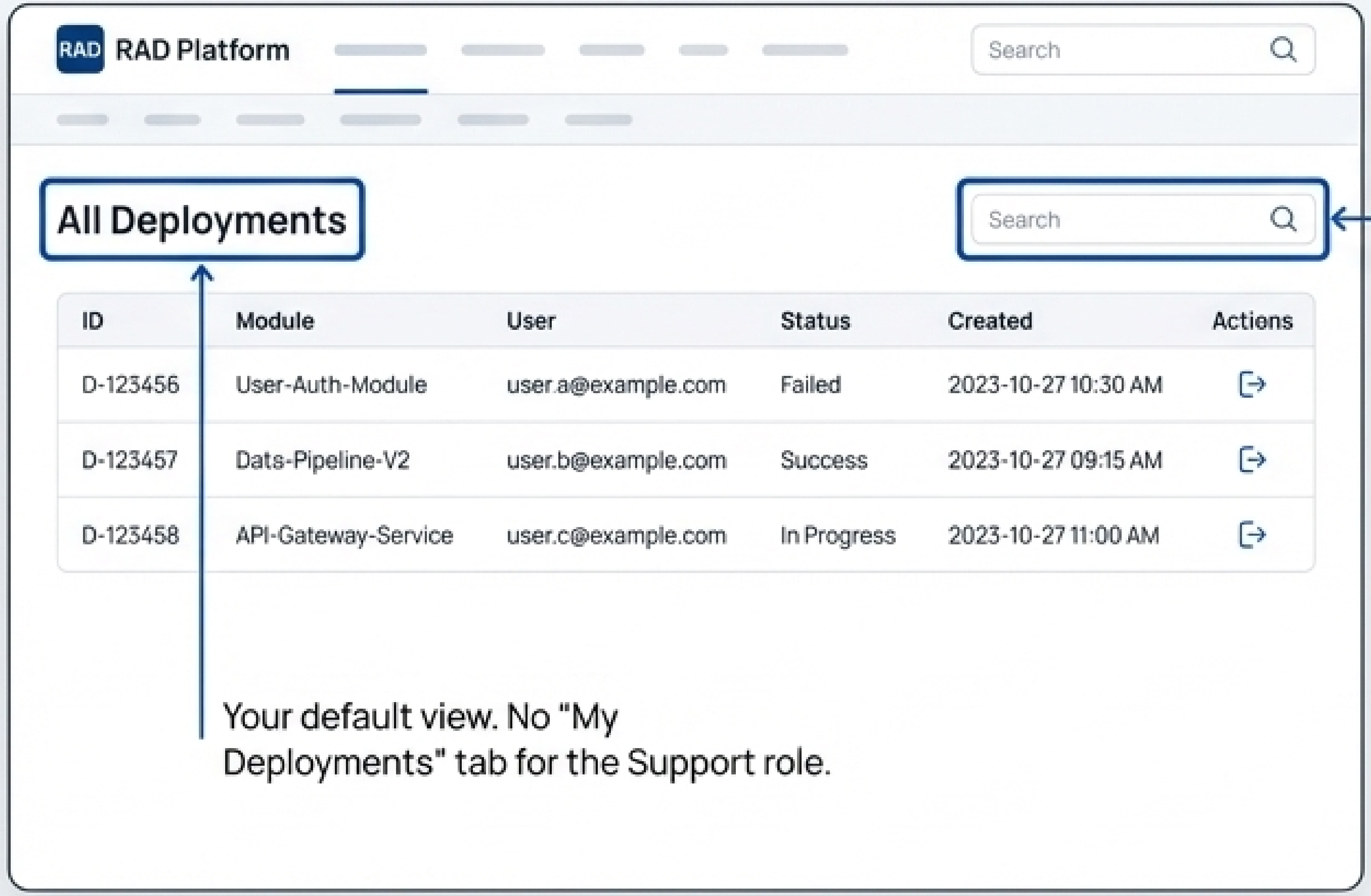
The Path to Resolution: A 5-Step Workflow

This is the proven, systematic process for handling any user-reported deployment issue, from the initial alert to the final resolution.

1. **Identify**
   Get the Deployment ID from the user.
2. **Locate**
   Find the deployment in the 'All Deployments' list.
3. **Assess**
   Check the deployment's Status.
4. **Investigate**
   Dive into the Logs to find the root cause.
5. **Advise**
   Guide the user to a solution.

Steps 1 & 2: Identify and Locate the Issue

Your investigation begins in your command center: the **Deployments** page. Your default “All Deployments’ view is the starting point for every case.



Search across all deployments.

Key Actions

Action 1: Get the Deployment ID.

This is the fastest path to a specific issue.

Action 2: Use the Search Bar.

You can filter by:

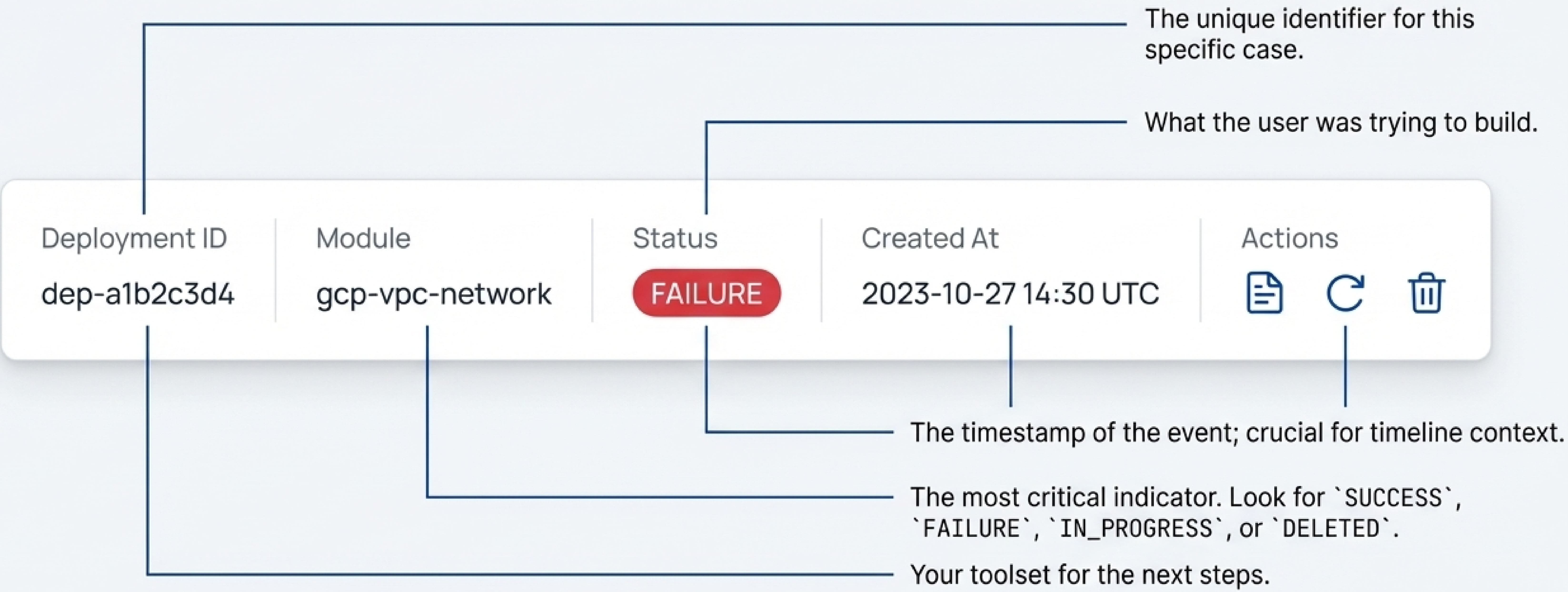
- Deployment ID
- User Email
- Module Name

Pro Tip

Searching by User Email gives you the full context of their recent activity and all associated deployments.

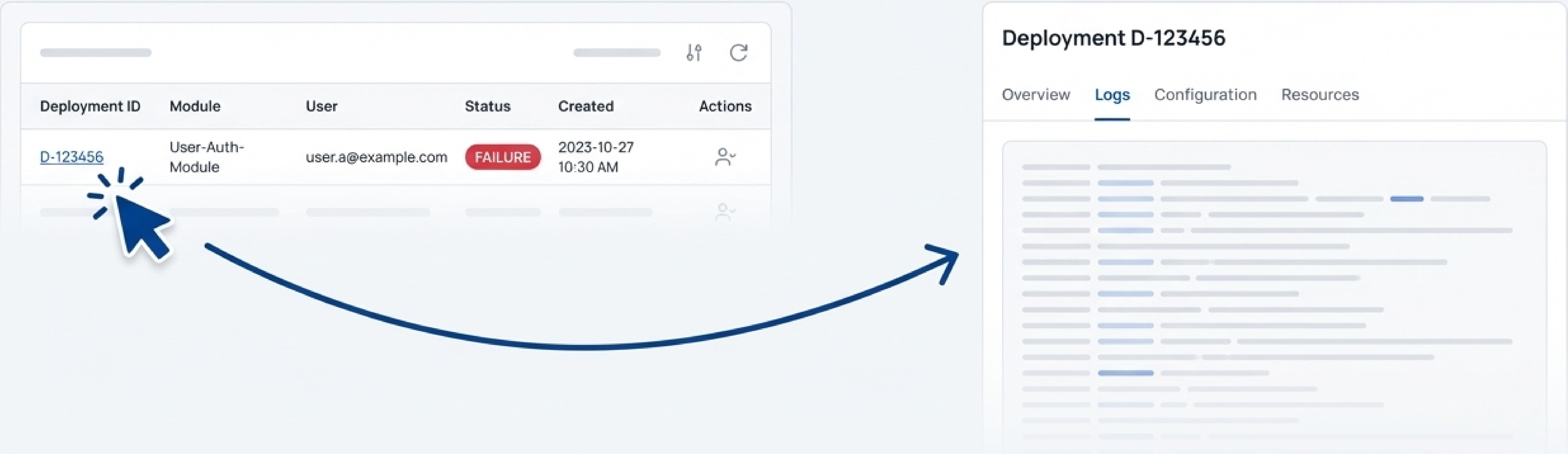
Reading the Vitals: Key Information at a Glance

Before you click, each row in the deployment list provides critical, high-level information. Learn to read these signs to quickly assess the situation.



Step 3: Assess the Status and Begin Investigation

A `FAILURE` status is your signal to go deeper. Clicking on the Deployment ID opens the full story, including the build logs where the root cause is found.



Step 4: Finding the Root Cause in the Logs

```
[INFO] Starting Terraform build process...
[INFO] Initializing modules...
[INFO] Successfully initialized
[INFO] Planning infrastructure changes...
[WARN] Deprecated resource usage detected
[INFO] Plan generated. Applying...
[ERROR] Error applying plan:

Error: Invalid variable "project_id" - The project ID
does not exist or you do not have permission to access
it.

[INFO] Build failed with exit code 1
[INFO] See above for error details.
[END] Build process terminated
```

The **Logs** tab contains the raw output from the Cloud Build and Terraform processes. This is your ground truth for debugging and where you will find the specific error message.

Common Errors to Look For:

- Permissions denied
- Invalid variable
- API quota exceeded

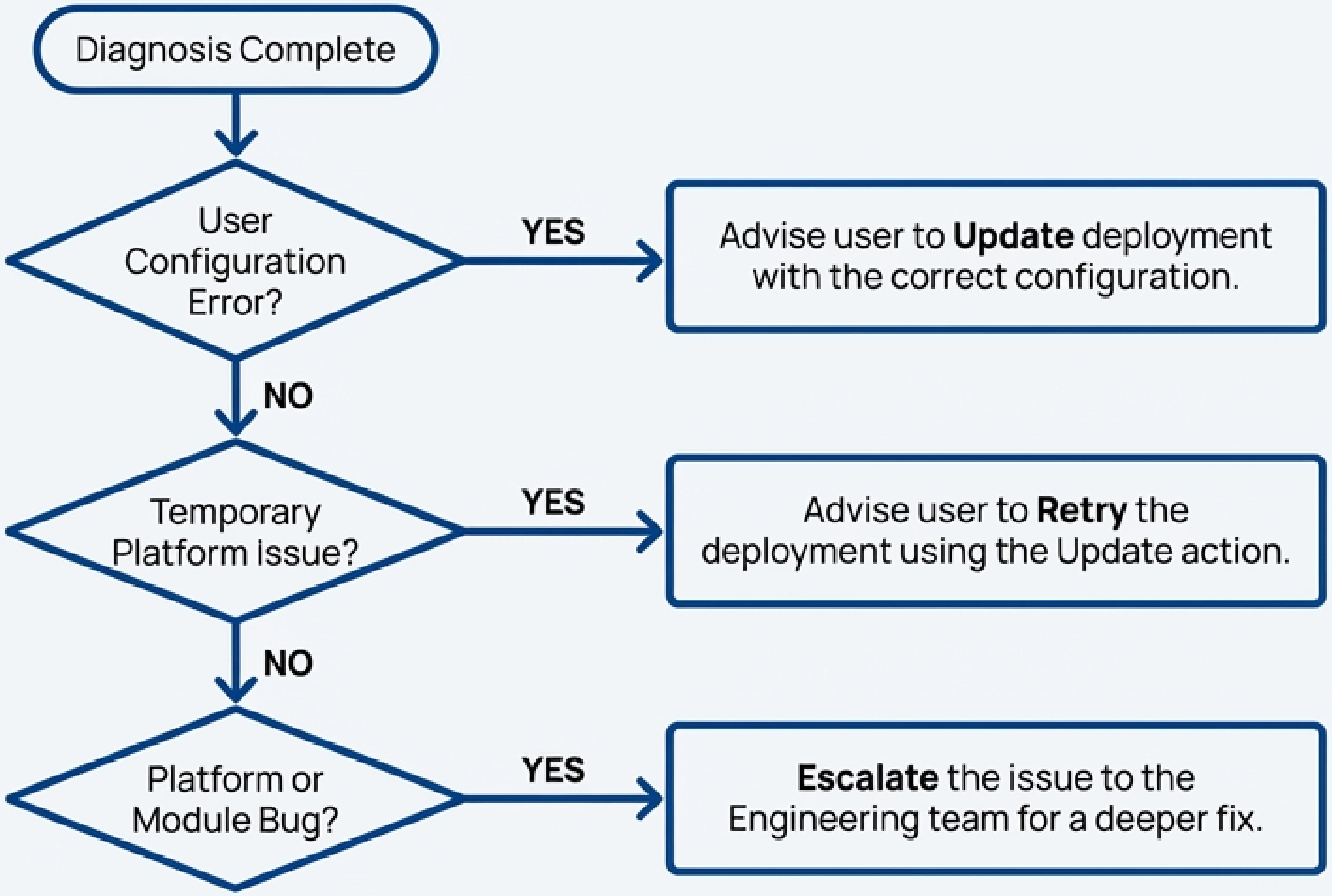
Additional Context:

Check other tabs for supplementary information:

- **Variables:** See the exact inputs the user provided.
- **Resources:** View the list of cloud resources created by the deployment.

Step 5: Advise and Empower the User

Once you've diagnosed the issue, your role is to provide clear, actionable advice to the user, guiding them toward a successful resolution.



The Support Action Toolkit

You and the user have several powerful actions available on a deployment. Understand what each one does and when to use it.

Update

Retries a failed deployment with the same configuration. Useful for temporary issues or after a user has fixed an external dependency.

Delete

Destroys the cloud infrastructure by triggering a `terraform destroy` process. This action is destructive and cannot be undone.

Purge

Removes the deployment record from the RAD Platform but does not delete the underlying cloud infrastructure. Use with caution.

Rate

Allows users to rate a module from 1-5 stars. Encourage this to help improve the module ecosystem.

Going Above and Beyond: Publishing a Fix

Because you have full Partner privileges, you can directly contribute to fixing problematic modules or testing updates before they are released to users.

1

Configure

Navigate to your **Profile** to set up your GitHub token and connect your module repository.

2

Publish

Use the **Publish** page to scan your repository and publish new versions or updates to existing modules.

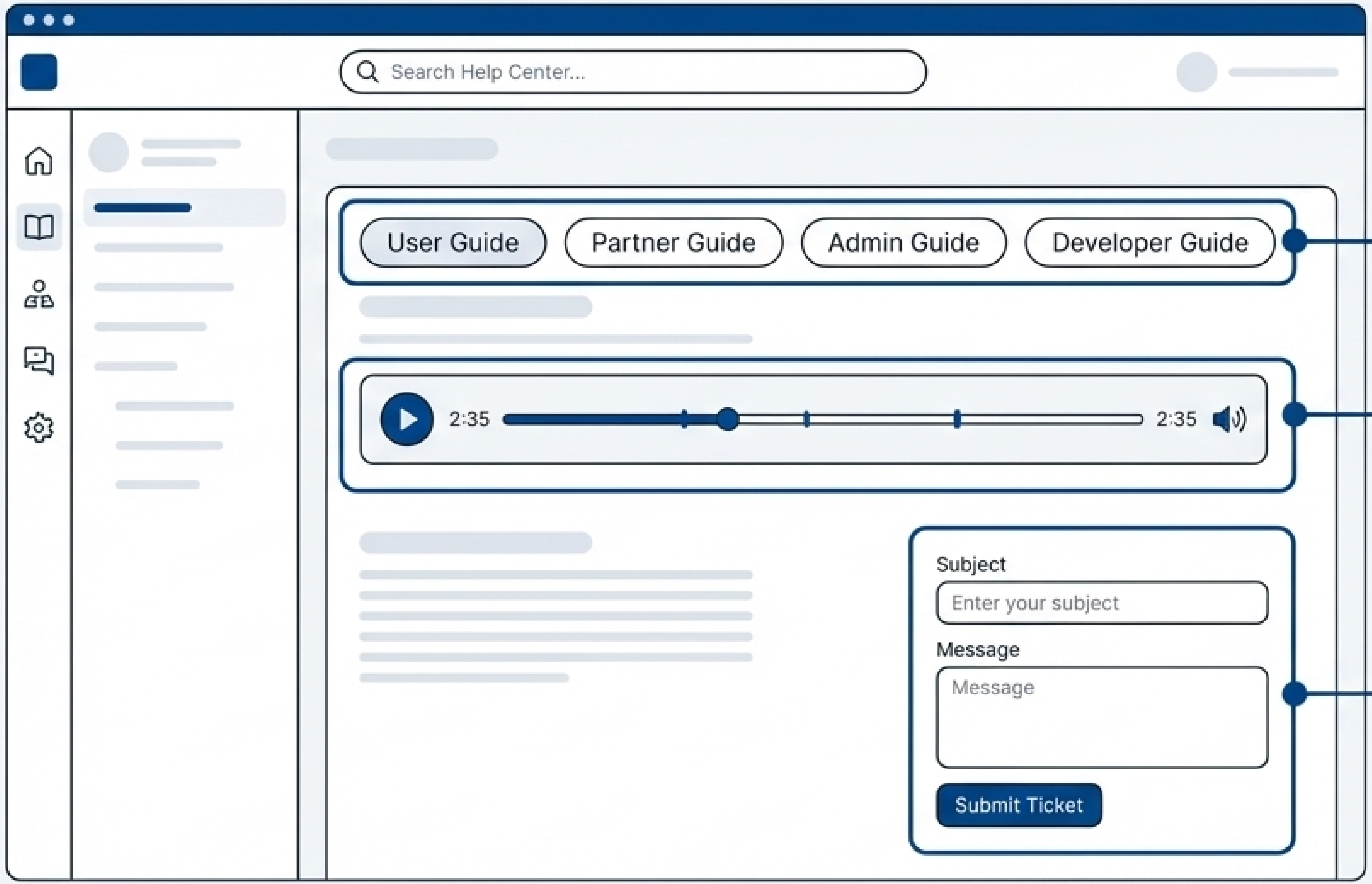
3

Collaborate

Use the AI agent, 'Jules,' to help debug module code or suggest improvements before publishing your fix.

Your Knowledge Base: The Help Center

When you or a user need more information, the **Help** page is your central resource for documentation, tutorials, and support channels.



Documentation Guides: Role-specific guides for users, partners, and admins are always available.

Audio Tutorials: Guided audio walkthroughs are available for key features and complex workflows.

Contact Support: The official channel for users to submit tickets and reach your team.

Know Your Escalation Path: Admin Capabilities

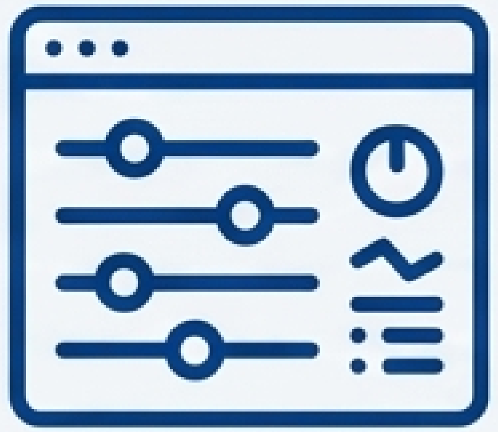
For platform-wide issues beyond a single deployment, you will escalate to a Platform Admin. It is useful to know what tools they have at their disposal.

User Management

Admins can manage all user roles and permissions across the entire platform.

Send Message

They have a tool to broadcast notifications to all users or send direct messages to specific groups.



Platform Configuration

They have full control over billing, global platform settings, and third-party integrations.

Your Support Playbook: A Recap

Remember these **core principles** to effectively diagnose and resolve any user deployment issue on the RAD Platform.

- Your power is the ability to view **All Deployments**.
- The **Deployment ID** is your key to every investigation.
- The **Logs** hold the answer to “what went wrong?”
- Your goal is to **Diagnose and Advise** the user.
- Leverage the **Help Center** for documentation for you and your users.