



# Honeybadger Moves from Self-Hosted Postgres to Crunchy Bridge



Honeybadger is an application monitoring service for keeping track of uptime, errors, metrics, and even logs. Honeybadger has always enjoyed open source tools like Ruby on Rails and Postgres and has developed pretty deep internal Postgres knowledge along the way. They had been self-hosting Postgres on Amazon EC2, using open source Patroni to manage high availability and pgPartman to manage automatic partitions. So why does someone who knows this much about Postgres need help? Because scaling is very hard.

Honeybadger has a small team and they no longer wanted to rely on the knowledge of a few key team members. “We wanted a better approach to emergency responsiveness,” said Honeybadger Co-Founder Ben Curtis.

While the Postgres database itself was working for them, it was too difficult to scale their internal expertise. With their Co-Founder Ben Curtis shouldering a lot of the Postgres burden personally, they began searching for additional support.

## Real Postgres expertise

Honeybadger wanted a vendor that could provide real Postgres knowledge. They didn’t want to move to a system where they would have escalations and tiers, waiting days to get a real answer to a problem or find an actual expert. “Postgres is central to everything we do, so picking a good team is imperative,” said Ben Curtis. Ben said that they didn’t think RDS would be a good fit, and that they would not get enough help and potentially end up paying more for their Postgres services, while continuing to shoulder most of the burden for managing the system.

“I felt like if we went with RDS, we’d still be on the hook for doing most of the heavy lifting. I know Craig and the team behind Crunchy Bridge. They really know Postgres. There wasn’t a compelling reason to even look at any other vendors.”

**Ben Curtis, Co-Founder at Honeybadger**

Honeybadger has been really happy with the support at Crunchy Bridge since joining. The Bridge support team is entirely staffed by Postgres experts, who take a deeper look at the issues, rather than firing off a link to documentation or assuming it is a user or application error.

“Support has been really good. We get prompt responses. And it is always an educated response. This kind of support is lacking in so many scenarios nowadays. To get a good response from someone that knows what they’re talking about is really valuable to us.”

## Ben Curtis

Co-Founder  
Honeybadger



**crunchydata**  
crunchydata.com



# Honeybadger Moves from Self-Hosted Postgres to Crunchy Bridge



“Support has been really good. We get prompt responses. And it is always an educated response. This kind of support is lacking in so many scenarios nowadays. To get a good response from someone that knows what they’re talking about is really valuable to us.”

**Ben Curtis, Co-Founder at Honeybadger**

A couple months ago Honeybadger had some serious performance issues crop up all of a sudden. They were able to get on a call with someone from the Bridge support team and get help right away. “I’ve been really happy that when we’ve needed it, help isn’t too far away.” said Curtis.

## Less worry

Honeybadger knew that when moving to hosted Postgres they’d lose SSH access to their machines and some of the direct control they’d previously had over their databases. For some developers who’ve self-hosted Postgres for years, losing access can be a worry. But Honeybader was excited to let go and have less access.

“Letting go of the database access wasn’t hard, it was something we actively wanted to do. We wanted someone else to be there in an emergency. I was excited to hand that over to someone else.”

**Ben Curtis, Co-Founder at Honeybadger**

Honeybadger was excited to hand over the reins of database management to Crunchy Bridge. This meant there were no more middle of the night alarms going off when something was down. Everyone on the team could take longer vacations away from their laptops. Honeybadger is getting to enjoy all the benefits of using Postgres without the headache of managing it with a small team.

Another huge benefit for Honeybadger moving to Crunchy Bridge is a simpler security story. They are SOC II certified, so working with Crunchy Bridge means they have less for their own compliance. Crunchy Bridge itself is SOC II certified, and since Honeybadger’s team was removed from direct access, their compliance process has been simplified.

“I felt like if we went with RDS, we’d still be on the hook for doing most of the heavy lifting. I know Craig and the team behind Crunchy Bridge. They really know Postgres. There wasn’t a compelling reason to even look at any other vendors.”

**Ben Curtis**

Co-Founder  
Honeybadger



**crunchydata**  
crunchydata.com



# Honeybadger Moves from Self-Hosted Postgres to Crunchy Bridge



## Time savings

When you self-host you're paying the lowest possible infrastructure cost, but that's only part of the equation. You don't always account for your time and effort of maintenance. Honeybadger went into this move prepared to pay more, knowing they'd get the benefit of having better support and spending less time on routine database maintenance.

"Moving to hosted Postgres and adding some costs was an easy sell internally. When I told everyone they could stop calling me and this was backed by a whole team and company totally focused on Postgres, they immediately saw the value."

**Ben Curtis, Co-Founder at Honeybadger**

The other big thing to consider for Honeybader in terms of cost management was the time they were spending on minor and major Postgres upgrades.

Honeybadger was committed to staying on the modern version of Postgres every year and adding security patches as they came out quarterly. They had a pretty good cadence of testing, creating new images, scheduling downtime, working with their own internal notifications, and manning the cutovers. But of course, this took their team quite a bit of time several times a year.

"Since moving to Crunchy Bridge I've spent a lot less time on Postgres maintenance, primarily the upgrades. I'm able to worry a lot less."

**Ben Curtis, Co-Founder at Honeybadger**

With the time savings, better security story, and added support, Honeybadger has made the transition from self-hosted Postgres to Hosted Postgres and they couldn't be happier about it.

"Letting go of the database access wasn't hard, it was something we actively wanted to do. We wanted someone else to be there in an emergency. I was excited to hand that over to someone else."

**Ben Curtis**

*Co-Founder*  
Honeybadger



**crunchydata**

[crunchydata.com](https://crunchydata.com)