

A CASE STUDY BY



FOR



Elastic is a search company helping organizations of all sizes use the power of search to solve critical business challenges. Its technology is used across application search, logging, application performance monitoring, and security analytics. In the early days of Elastic, the core of what Elastic offered were its open source products — [Elasticsearch Service](#), Logstash, and Kibana — where many answers to questions were available on the web and through the community.

“You can search the open web and find wealth of information,” said Marty Messer, vice president of customer care. “In order to support our early customers, we had to explain that they might find 50 answers, but that our support engineers we could help them decide which one was best for their particular use case.”

The products that Elastic offers has grown to include hosted and managed solutions. For organizations who want a turnkey product to manage their Elastic Stack environments, Elastic provides [Elastic Cloud Enterprise](#), and for users who want to support their Kubernetes deployments, Elastic offers [Elastic Cloud on Kubernetes](#).

This includes many free and paid proprietary features and solutions, as well as, full paid subscription offerings like Elastic Cloud Enterprise (ECE). “Search for answers about ECE and you won’t find much, because customers keep the details of their environments private, and each installation is relatively unique to the underlying architecture,” says Messer. “While it is built using all of our open source products and based on our Elastic Cloud service, it is full of proprietary features common in enterprise software. So we have to build and curate all that knowledge in house. It was a real shift for us.”

Messer and others quickly realized that the organization’s traditional approach to customer support and internal documentation would have to change.



You have knowledge stuck in very few people’s heads, the people who built it. To extract it is very difficult. So that’s the real driver. That’s really where it all started.

Marty Messer | Vice President, Customer Care

Let your team use the tools they already know

When Elastic was a relatively small organization and most of its products were open source, customer support could be handled through email and group chat. But over the last few years, it outgrew those methods. “When somebody has a question from a customer or is in the middle of an issue, we’ve gotten to a size where it’s impossible just to go to chat and ask the question,” says Messer. “You can do it, and everyone on the team will try to be super helpful to each other, but you miss it. It’s just way too synchronous, and half the time it scrolls away before you have a chance to read it. We wanted something more persistent, asynchronous, and global.”

At first, the Elastic team turned to the internal tools it was familiar with. “We already use GitHub for our code,” says Messer. “We also use it anytime that we need to escalate an issue to our dev teams from our customers. We figured we would try using that, because that’s where our engineers spend their days.” The support team began to put questions and answers into a GitHub repo created specifically to store this kind of knowledge.

Unfortunately, that solution made it difficult to search for answers later on, and there was no way to identify the best answer if multiple people had contributed ideas for possible solutions. Leaf Lin is a support engineer and cloud tech lead, a role that involves supporting both external customers and internal teams. “We used Github issue tracking, but the search wasn’t very



flexible. It also was not easy to find which answer is the best based on likes, you had to scrub through a ton of information to find the valuable stuff,” says Lin.



We also tried using a Confluence wiki and Salesforce knowledge base articles. There was a voting system but people can't find stuff as easily or identify which answer was accepted as the best solution by our team.

Leaf Lin | Support Engineer and Cloud Tech Lead

In the end, the team tried exporting everything they had written across these three knowledge base platforms into Elasticsearch, where they built a huge database to act as a centralized wiki. “It worked, to a degree, but it was read-only, there was no good way to write,” says Lin. “And even if you find an answer you like, you can't upvote it.”

Eventually, the support team realized there was another tool all their engineers were familiar with and using on a day to day basis. “Stack Overflow is the site we visit most often for support related type questions, so it made obvious sense that we might have better luck internally with something built for exactly the thing we're trying to do,” says Messer.

Helping support find answers quickly

Elastic Cloud customers have access to support as part of their subscription. The support team is the first line of defense, but when they can't handle something, they escalate it to the development team which includes the core developers who author features for Elasticsearch, Kibana, Beats and Logstash.

“When I came into Cloud, we were struggling with that, the process of escalating efficiently from support to development for our Cloud products,” says Suyog Rao, Director of Engineering at Elastic Cloud. There was a lot of buzz around Cloud, and it was new and growing rapidly. “We were at a point where the same questions were being asked over and over, and it was not very structured. It was all on Slack, it was through email, through the wiki. There were multiple sources of information that we had to deal with.”

When support staff couldn't find an answer, they would reach out to the developers who built the product. “Developers at Elastic love collaborating with support, but constant context switching because you're being asked similar questions by multiple people across multiple sources was consuming a lot of time. That's what we were trying to solve,” says Rao.

Rao stresses that this isn't some division being drawn between the two departments. “Development and Support should work hand in hand. It's not that developers are too busy to assist on support.” But the only way engineering can help at scale is to have some kind of structure where they can capture some of the solutions to existing issues and common patterns.



Engineers should help solve the hardest questions, the unknowns, where being familiar with how the product was built is essential. But we don't want to keep answering solved problems over and over again. That's where Stack Overflow really helps.

Suyog Rao | Director of Engineering, Elastic Cloud

For example, a common problem among customers who were scaling quickly was that they would under-provision their cluster. You start small, and then send a lot of data into that, and that would manifest into multiple problems: “I can't upgrade” - “I can't back up my data” - “I can't add

any new visualization to my data.” The crux of the problem is capacity, but it may not always be clear that customers with varied symptoms are suffering from the same illness.

With Stack Overflow, it was easier to find the answers to existing questions and to organize knowledge around certain topics using tags. “We found a way to show the support team, here’s the base problem that is common across multiple customers, and here is the area where you can learn all about the various solutions we’ve implemented over the years,” says Rao. “Once they know that, they don’t even need to involve engineers.”

At the same time, Rao has noticed that the work the support team does on Stack Overflow creates useful data the development team can tap into as they decide what to build next. Folks from across the development team, not just the support team, subscribe to an email digest that catalogs the top questions and answers. “Not everyone gets into the Stack Overflow system day to day, but when we receive the email updates, we channel that into our roadmap planning, creating a feedback loop.”

A way to transfer soft skills - institutional knowledge about client relationships

Working with large enterprise clients requires a lot of finesse, especially when they will be relying on you for mission critical infrastructure. “We tend to work with customers on their architecture and how they scale, more consultative topics. That often means there’s no simple answer,” says Messer. “It’s not so much, ‘What does the setting do?’ We already have good docs on that. It’s really about, I’ve got a customer who wants to scale from 10 terabytes to 10 petabytes. How should I think about approaching that conversation with them?”

Despite this dynamic, Elastic has resisted the idea of siloing off certain engineers for different clients or topics. “We’re not the kind of place where we have a team that comes in over top and says, ‘We’re here, we’re the scaling experts.’ Every engineer’s expected to have that conversation with a customer if they get asked.”

The dynamic nature of Stack Overflow, where anyone can ask or answer a question, where anyone can vote, and where anyone can comment, helps to preserve a useful, well-organized version of these larger, more nuanced conversations. “The way one of our engineers needs to approach thinking about having that kind of conversation with a customer, that’s one of the big pieces that gets bubbled up.” To be able to tag and organize things, to make clear what answers are fresh and which ones are not, has been invaluable. “Sometimes our answer for a customer is locked and loaded, sometimes it’s more experimental. To be able to get that disseminated across my team without having to do a synchronous video call each time, that’s a big advantage.”

Traditional documentation is great for explaining what all the features and functions of a product do and all the possibilities of a configuration. But it doesn’t help you navigate the real world problems customers are having. “Docs are lots and lots of information. But only when you intersect it with what that customer’s trying to do does it really become knowledge,” says Messer. **The ability to rank, edit, organize, and comment on questions and answers in Stack Overflow makes it far easier to organize, archive, and search through this kind of knowledge.** “You need both, because it’s the use case, it’s the context that information is put in, that’s when it becomes really valuable.”

