



Implementing Search for Your Knowledge Base

Implementing powerful search for your knowledge base is easy to understand in theory, but can be intimidating in practice. This ebook walks through the steps required, outlines available options, and provides guidelines to ensure smooth implementation.



READING TIME: 20 MINUTES



elastic



Table of contents

Introduction3

Build vs. buy: Which implementation makes sense for you?.....4

 Out-of-the box search5

 Cloud-based search.....6

Next steps: The four phases of implementing search7

Planning: Designing search and establishing timelines9

Implementation: Realizing your vision11

Testing: Qualifying and quantifying search quality13

Maintenance: Long-term optimization and operations15

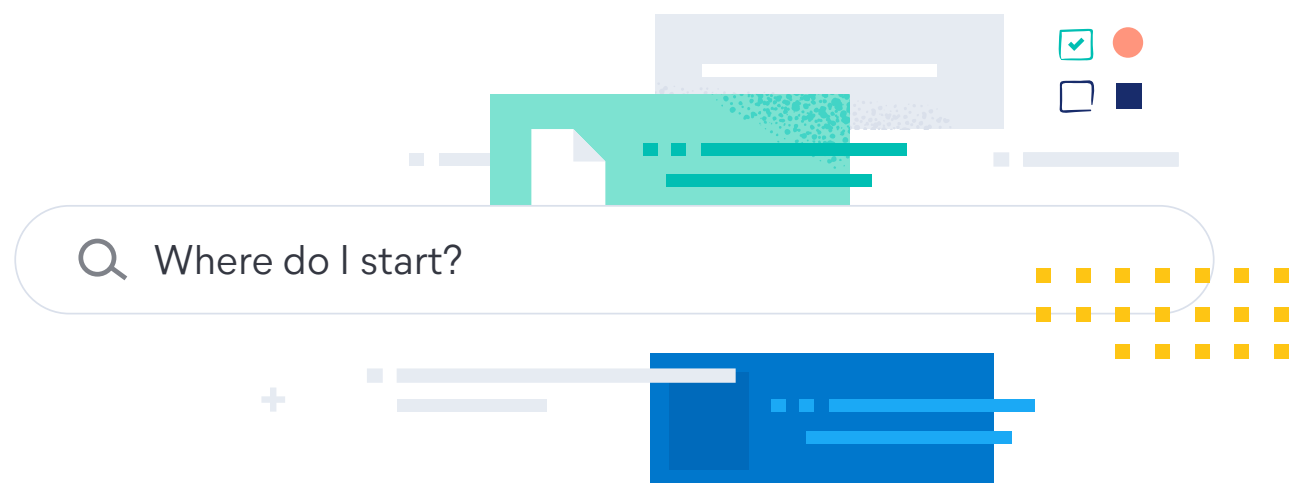
Introduction

Congratulations on making the decision to implement search for your knowledge base! You're likely already aware of the benefits of having effective search, such as fewer inbound support tickets, streamlined customer support, and a better overall customer experience.

You may even have seen stats like this: 45 percent of companies offering web and mobile self-service reported an increase in site traffic and reduced phone inquiries.

However, when it comes time to implement your search, you may encounter a whole set of issues you've never had to deal with before. Fortunately, you've come to the right place for help. In this ebook we'll talk about the different search implementation options available for your knowledge base and take you through the four phases of search implementation:

1. Planning
2. Implementation
3. Testing
4. Maintenance



Build vs. buy

Which implementation is right for you?

The type of search implementation you choose will depend on the goals you've defined for your search, the resources you have available, and your unique working environment.

Let's take a quick look at two different types of search implementation: out-of-the-box content management system (CMS) search and cloud-based search. The basic differences are outlined below, with more in-depth explanation on the pages that follow.

Out of the box

- No work required
- One size fits all
- Cannot be customized
- Weak text analysis and relevance
- Unable to customize search results
- Weak analytics
-

Cloud search

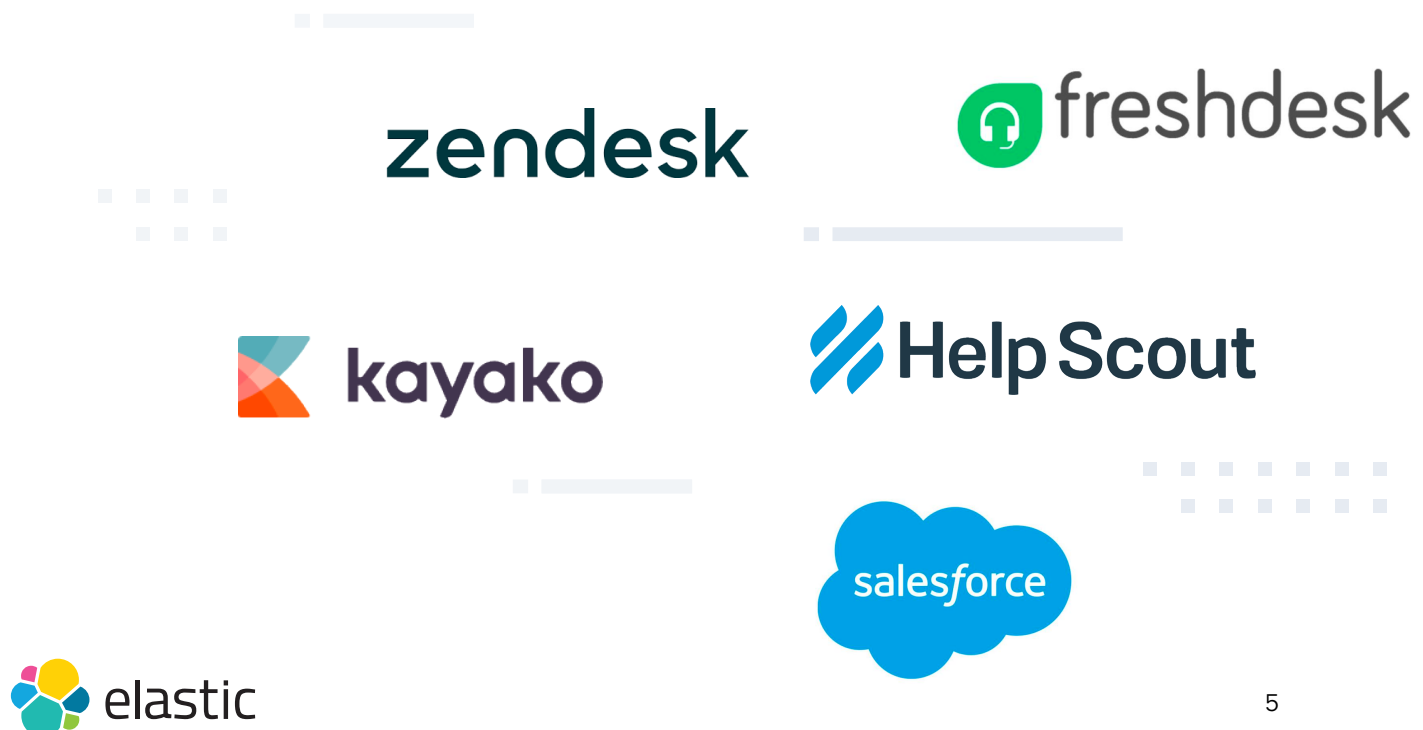
- Faster implementation
- No ongoing maintenance costs
- Built-in relevance and language recognition technology
- Customizable via dashboard
- Analytics dashboard built into product

Build vs. buy

Out-of-the-box CMS

During the planning process, some companies may feel that the out-of-the-box search built into their CMS is adequate for their customers and support agents. While in some cases this may be true, many companies soon discover that out-of-the-box search is inherently inflexible and cannot be customized for the specific needs or desires of your support team.

As a one-size-fits-all solution, CMS search relevance models cannot be changed, leaving support teams guessing about how they might influence which search results appear at the top of the list for a given query. Perhaps more importantly, out-of-the-box search is often built with little to no support for common language problems (such as spelling correction, synonym recognition, or basic language parsing technology). Companies that want a customized search solution with more advanced relevance and language modeling technology will need to invest in building new search on their own or look to a third party to build and host search.

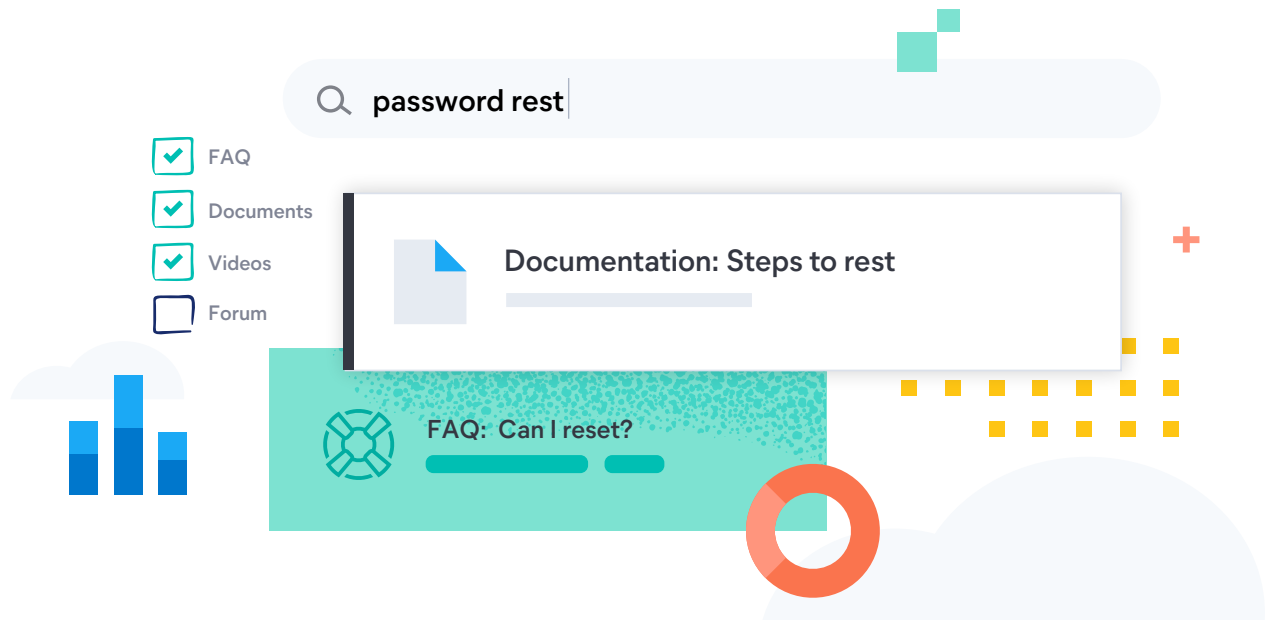


Build vs. buy

Cloud search

If you want the benefits of an advanced search engine and language modeling technology but don't have the engineering time to build and subsequently maintain search, then cloud-based search is probably the best option for you.

Cloud search solutions allow complete control over your search experience, but free your engineers up from building advanced relevance models or developing natural language processing for your search engine. Hosted search also saves your engineers from dealing with recurring software updates and ongoing server maintenance, and often comes with advanced analytics and optimization dashboards.



Next steps

The four phases of implementing search

Now that you understand which implementation options are available, it's time to learn more about the four phases of search implementation: planning, implementation, testing, and maintenance. Below are the basic elements of each stage, with more in-depth explanation of each on the pages that follow.



Planning

Gather stakeholders, design search, set goals, propose timeline



Implementation

Indexing, frontend coding, backend server configuration



Testing

Check search result quality, test common language problems



Maintenance

Optimize search results, maintain servers and software

Planning

What is the goal?

The first step in the planning phase is to look at your own people, processes, and systems and develop a set of best practices for your own knowledge base to ensure a smooth implementation. At the end of planning, you should know how your new search will work, who will be responsible for executing what, and how long the project will take.

Who is involved?

Once these are in place, you can move on to gathering your project team members. Resources you'll need to involve include:

- VP of support to champion and guide
- Software engineer for development work
- UX designer for frontend work
- Customer service representatives and support content creation team to provide input on requirements

Planning

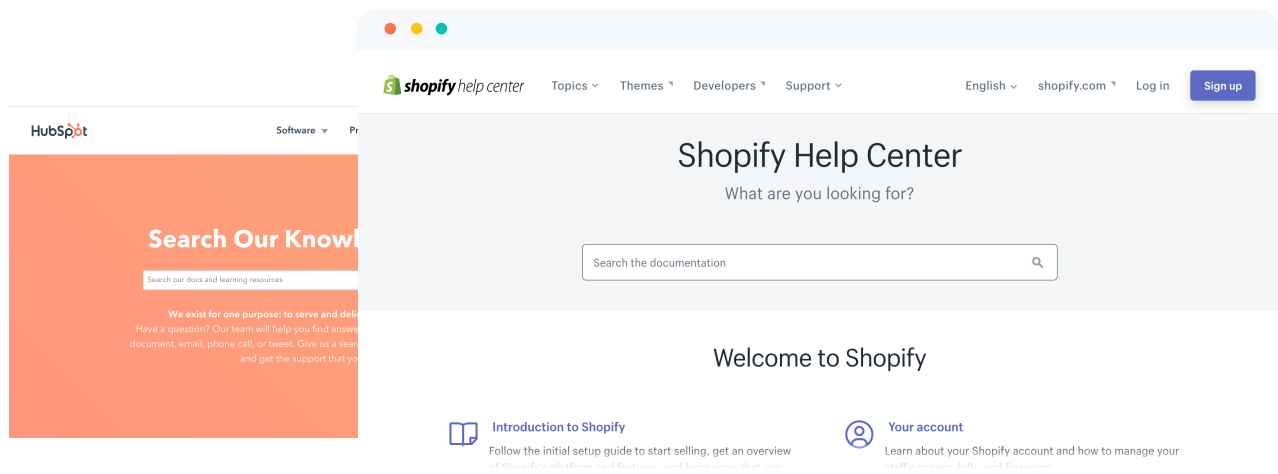
How will search change?

With your team in place, you can move on to one of the most important steps in the planning process — deciding how you want your search to work. Why is this step so critical? Because what you define in this step determines how visitors will experience your search when they visit your knowledge base.

For example, you might want customers to be able to search by topic, content type, or date/time frame, as well as have the ability to filter results by topic, tags (such as “getting started” or “troubleshooting”), content type, or reviews.

To help customers find your content faster, you might also want to implement an autocomplete experience that features results rather than suggested queries. This removes a step in the search process and immediately gets visitors to their destination.

Once you’ve determined how you want your search to work, it’s time to think about how you’ll index your knowledge base content — a topic discussed in depth in the implementation section.



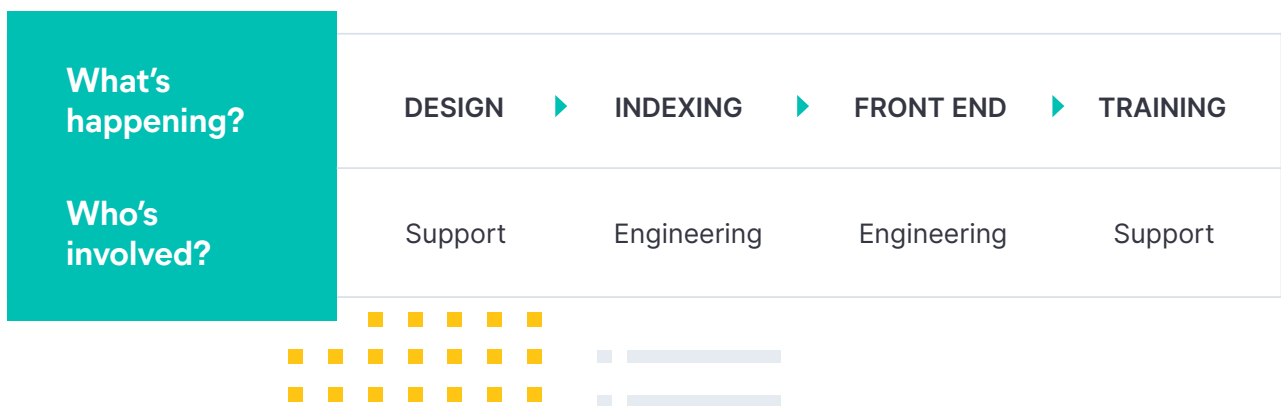
Planning

Who will do what, and when?

In the last step of the planning process, you'll estimate project timelines, as well as develop training and communication schedules. For the overall project timeline, expect to allocate between one week for a very simple paid solution to several months for an open source search solution.

You'll also want to allocate time for training your customer service representatives and your support content creation team so they can understand how customers will be searching and the process they'll need to follow when adding new content to your knowledge base.

Finally, you'll need to develop a communications strategy in order to let people in your company know that the new search is coming, and how much easier it will be for them and their customers to quickly find the information they need.



Implementation

How can I index my website?

Data feed: Establish an XML, CSV, or RSS file transfer protocol

If you're working with a third-party provider, they may ask you to pass on all of the information you want to index via a data feed. This process involves setting up an FTP transfer of your information in XML, CSV, or RSS format every 24 hours.

Aside from being cumbersome to configure and maintain, data feeds often leave your search index out of sync with your live website offerings, since website changes are only communicated to the provider when the FTP transfer goes through.

API integration: Build database connection between website and search engine

An API will accomplish the same as a data feed, but without the annoying lag — information is passed on instantaneously to your search engine. API integrations also allow complete control over your engine schema, which allows you to more precisely control how your search experience will work. If someone on your team is experienced with API integrations, then this might be the way to go.

Web crawler: Website information automatically extracted from HTML

If you don't want the hassle of dealing with a full-scale API integration, or you don't have the technical expertise to do so, you can get the same results using a web crawler.

When working with a web crawler, you simply provide a URL and the web crawler automatically extracts website information from the HTML. In the case of Elastic App Search, you can control exactly what information is extracted by specifying website information with custom Elastic `<meta>` tags, which speak directly to the web crawler and allow you to create a custom search engine index.

Implementation

Do I need to buy servers?

Once you've indexed your content, you're ready for the next step in the implementation phase: resolving server and operational issues.

There are two reasons to monitor your servers if you self-manage your index:

1. To ensure that you have enough servers to handle traffic
2. To resolve any speed issues that might arise from customers searching your knowledge base in other parts of the world

One of the many advantages of cloud-based search is that these issues are entirely handled by your provider. On the other hand, if you're building your own search, you'll need to buy servers in data centers around the world to serve international audiences at speed. Additionally, you'll need to ensure you have the correct amount of servers to handle your fluctuating search engine and website traffic.

How long will the frontend work take?

In this step, you'll take what you've planned and work with designers and frontend engineers to put everything in place for the user interface.

Allocate at least one week of work time for designers and frontend engineers to put this into place. As with all phases, be sure to add in some buffer, as tasks often end up taking longer than anticipated.

Testing

How do search results look for my most important queries?

In order to ensure your search is working properly, you should run tests of how your search handles your top 50-100 queries. The key here is analyzing relevance and asking yourself: Do these search results match my expectations? Are these the most helpful answers for this search? If not, you should customize the results for these queries until they match expectations.

How does my search handle common language issues?

In addition to relevance, your team should investigate how your search handles common language issues such as misspellings and phrase matching. Below are some of the most common situations to test.

Misspellings ▶

corrects “templæt” to “template”

Bigrams ▶

recognizes equivalence of “e mail” and “email”

Dashes & spacing ▶

recognizes equivalence of “i phone” and “iphone”

Stemming ▶

recognizes equivalence of “starter” and “started”

Phrase matching ▶

distinguishes between “iOS 9 apps” and iOS 8 apps”

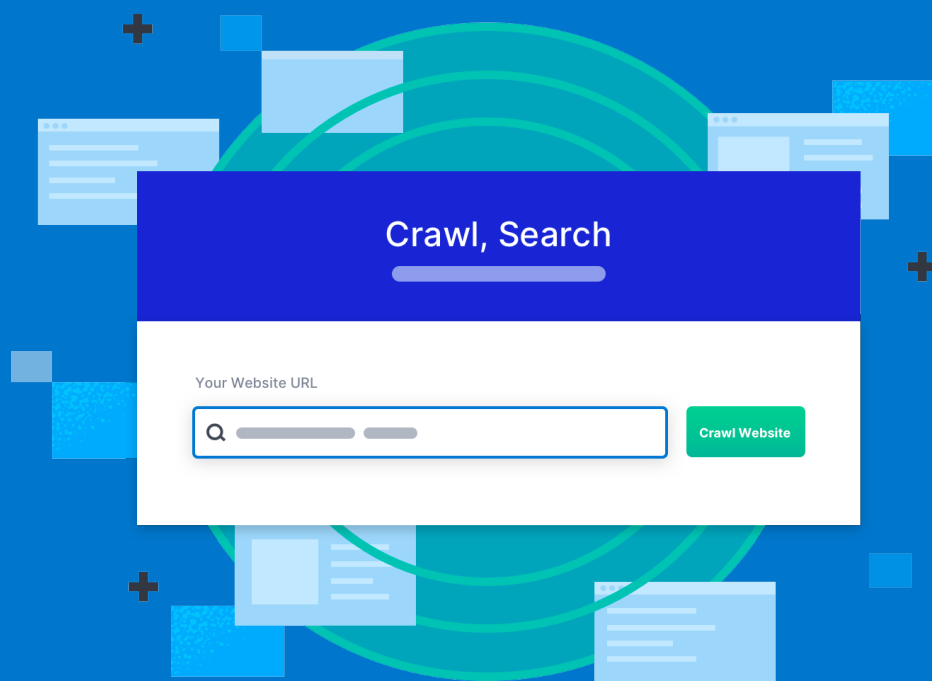
Maintenance

What is required to maintain my index structure?

Once you've determined how your index will be structured, you'll need to have every piece of content created moving forward conform to that indexing structure.

If you have an open source solution, stay on top of updates by keeping in close touch with software developers. Keep in mind that one engineer could spend almost a full week making sure their search works with software updates.

If you're using a web crawler, maintenance will be a lot simpler. When you add <meta> tags to your content, changes are made on the template level. Once your site is recrawled, that content is automatically indexed. Finally, ensure that engineers have mapped your CMS to the correct indexing structure.



Maintenance

How can we optimize our search over time?

To optimize your search and case deflection rates over time, be sure to keep track of KPIs that matter most. These could be clickthrough rate, search exit rate, percentage of searches with no results, or other metrics. The key is not to get bogged down with metrics, but to choose carefully those that demonstrate the most impact and continually monitor how these affect your inbound ticket volume.

Finally, optimize your search by continuously customizing your search engine.

Tune your algorithm to control what impacts the order of search results.

FIELD

popularity

IMPACT

2.5

FIELD

average rating

IMPACT

5

Customize results for individual queries to feature the most important articles.

My **payments** are failing

Failed payments may be happening because...

Changing **payments** information

To change your payment information, log in to...

What **payments** are accepted?

We accept credit/debit cards, PayPal, bitcoin...

Can I make **payments** late?

Unfortunately, We cannot accept late payments...

About Elastic

Elastic is a search company. As the creators of the Elastic Stack (Elasticsearch, Kibana, Beats, and Logstash), Elastic builds self-managed and SaaS offerings that make data usable in real time and at scale for use cases like application search, site search, enterprise search, logging, APM, metrics, security, business analytics, and many more. Founded in 2012, Elastic is a distributed company with Elasticians working in countries around the world. Learn more at elastic.co.

