

Sources

- kubectldocs.kubernetes.io - Hugo + netlify?
- kustomize.io - Hugo? + netlify
- kubernetes-sigs.github.io/kustomize/ - Hugo? + GitHub Pages
- kubernetes.io - Hugo (docsy theme) + netlify
 - [kubectl reference](#)
 - [Declarative Management of Kubernetes Objects Using Kustomize](#)
 - [Managing Secrets using Kustomize](#)
- [Examples](#) - various markdown files not hosted anywhere outside of GitHub
- [Github README](#) - markdown
- [kustomise help topics](#) - markdown and built into kustomize.

Legend:

- **Keep this section, optionally with content** (see comments)
- **Remove this section, and optionally its contents** (see comments)
- **Unsure whether to keep or remove**

kubectldocs.kubernetes.io

- [Home page](#)
 - Top bar
 - [Installation](#) - links to a choice between kustomize or kubectl. Kustomize link links to a bunch of other links on different options for installing. Little frustrating to get to. I guess the Installation page can be a single page with multiple options since all the options are super simple single commands
 - [Guides](#) - links within links, split between kustomize and kubectl. I am often lost.
 - [Reference](#) - seems a lot better for kubectl than kustomize. I find the kustomize reference to be a bit too cluttered with examples. I always appreciate when reference pages stick to the base facts. The [kubernetes API reference](#) is a great example of this.
 - [FAQ](#) - the kustomize FAQ seems pretty nice here. Links to related github tickets is really helpful. Although not sure if the [Versioning Policy](#) belongs here?
 - [Blog](#) - hasn't been updated since 2019. This is the kind of thing which makes me think that a project is not being maintained.
 - [Contributing](#) - seems good
 - Language - only other option is Chinese and that seems very limited. Probably don't have the resources to maintain this right now.

- kubectl + kustomize front and center
- [Get started](#)
- [kubectl reference](#)
- [kustomize reference](#)
- External links
 - [Slack](#)
 - [SIG CLI Github](#)
 - [Kubernetes Twitter](#)
- [Guides](#)
 - Introduction
 - **kubectl** - NOPE
 - **kustomize** - YEP. Might just need a rewrite a bit and show the content of the files and kustomize output.
 - **Resources + Controllers Overview** - k8s stuff, not relevant for kustomize.
 - Configuration Management
 - **Introduction** - seems like a mishmash of kubectl and kustomize. Probably not worth migrating
 - **Apply** - mishmash of kubectl + kustomize again. Talks about recommending using the “-k” flag with kustomization.yaml instead of raw resources
 - **Secrets and ConfigMaps** - should be ported over and placed in a new section
 - **Container images** - seems irrelevant to kustomize
 - **Namespaces and Names** - talks about the namespace field in kustomization.yaml and briefly mentions nameprefix and namesuffix with no example. This stuff can be moved over, although I think it might be good to have a one-pager on these kustomize convenience functions
 - **Labels and annotations** - kustomize’s label and annotation features. Should be ported over
 - **Bespoke application** - seems like a very limited version of what is in the examples directory of the kustomize repo. Not sure if this is worth moving over if we have better examples. I think it’s probably also good to have a dedicated examples section with different scenarios, in addition to perhaps a simple example in a quick start or intro page.
 - **Off the shelf application** - I’m not sure how common the use case is for using kustomize with an off the shelf app, as Helm seems to be the defacto standard for those cases. I would suggest we use a solid example of an app which provides a kustomize config to deploy things.
 - **Kustomize components** - I find this example a little hard to follow because not all the files which comprise a component are explained, but I think I got there in the end. This should be ported but not too sure in which section it would belong yet. Concepts? With a more detailed example in an Examples section?
 - Resource Printing

- **Summaries** - just kubectl stuff
- **Raw** - kubectl stuff
- **Fields** - kubectl stuff
- **Queries and Options** - this contains the example of using kubectl get -k dir/ which is probably useful to people. I'm not sure where the integrated kubectl kustomize would go on this site (if at all) but this should be there.
- **Cluster Information** - kubectl stuff
- Container debugging
 - **Container Logs** - kubectl only
 - **Copying Container Files** - kubectl only
 - **Executing a command in a container** - kubectl only
 - **Port forward to Pods** - kubectl only
 - **Proxying traffic to services** - kubectl only
- App Deployment
 - **Diffing Local and Remote Resources** - mentions kubectl diff -k which is a very useful command. Belongs in the same place as the content in "Queries and Options".
 - **Access Multiple Clusters** - gives info on how to use kustomize with kubectl contexts. It's mostly kubectl stuff but I can see use cases here which may be helpful for certain workflows that we might want to highlight.
 - **Pushing Bases** - this seems super important and is very hidden. Definitely needs to be moved over and expanded upon.
- [Extending Kubectl](#)
 - **Plugin mechanism** - kubectl plugins stuff. I'm not sure what the difference between "kubectl apply -k" and "kubectl kustomize" is. Is the latter a kubectl plugin? And can be any version of kustomize? I find the integrated -k feature to be very opaque. If you can add kustomize as a plugin this way then I think it belongs on kustomize.io.
 - **Discovering plugins** - kubectl and krew stuff
- **Extending kustomize** - I find this page confusing. It should be ported but it definitely needs a bit of a rewrite to explain things better.
 - **Containerized KRM Functions** - lots of overlap with the top-level "Extending kustomize" page. I think this and the top-level page should consolidate the info on this and explain clearly, using a very simple example how this works. Expanded examples can be in the Examples section.
 - **Exec KRM functions** - keep this but perhaps move the guided example to the Examples page. The limited example before this seems to do a pretty good job of giving you an idea of how it works. Maybe we can somehow link relevant examples at the bottoms of these pages? Also for KRM functions and general kustomize examples? (Sidenote: the example proclaims how silly it is and how you would never use it. I'm pretty anti these kind of examples. Giving someone one practical example which they might actually use is always better IMO.)

- [Exec Plugins](#) - deprecated but I guess we still need it
 - [Go Plugins](#) - deprecated but I guess we still need it (perhaps we can combine this page with Exec Plugins as a kind of deprecated catch-all?)
- [Examples](#)
 - [Multibase](#) - looks good. Need to spend some time later just reading over all the examples. Editing, combining, identifying use cases and making a list of what to put up initially before we start writing anything brand new.
 - [Inline Patching](#) - seems like a lot of useful info. However, there was nowhere up to this point that explained how patching works. There are a [few sections in the Reference](#) which sort of act as both reference and introduction. I personally see value in structuring the docs so that you first see a dedicated explanation of patching behaviour with a very simple example, some use-case oriented and more advanced examples in the Examples section, and a very dry reference section which just has a minimal explanation of each aspect of the patches, with reference to the intro and examples.
- [Reference](#)
 - [Kubectl](#) - not going to list everything here because I don't think it'll be living on kustomize.io
 - Kustomize
 - [Kustomize Built-Ins](#) - this is a nice page. I would like to see some of these appear in the general docs as something like "convenience functions" or however they should be branded. They should also be sprinkled in throughout the examples.
 - [Commands](#) - seems like a section which is incomplete. Currently this is a separate page per command but I think it should be combined into a one-pager like the built-ins above
 - [Glossary](#) - love a good glossary! Something you just want to be like "wtf was this thing referring to again?" and control+f one a one-pager which gives you a brief description.
 - [The Kustomization File](#) - I feel like a lot of the text could be moved to the main docs section, rather than the reference section. Like I said above, I like my references dry and to the point.
 - [Bases, buildMetadata, etc](#) - again these are all links to individual pages which have examples on them. I think modeling things after the [kubernetes API reference](#) way of doing things is clearer. Of course it doesn't have to be exactly the same, the style of Kustomize Built-Ins is good. As long as we align on one consistent style internally.
- [FAQ](#)
 - [kubectl](#) - boo!
 - [kustomize](#) - we need this. I think it can be integrated into a left-hand menu along with the main documentation, rather than this conceptually whole different area of the website.

- [Blog](#) - RIP blog?
- [Contributing](#) - good to keep separate from the docs I think
 - [kubecti](#) - you again?
 - [kustomize](#) - haven't had the chance to read through this yet. Will do soon. First thoughts: I think the Community section should be split out to the top bar. It's very hidden here, and the community isn't just about contributing. Most of us are leeches!
 - [Documentation](#) - absolutely! But I guess most of this will have to be written from scratch

SyamSundarKirubakaran committed on Apr 29, 2021

kustomize.io

- I think we might be able to reuse some of the text on this page for the new landing page. Although the things it links to are very old and it would probably be good to update them to something more up-to-date. And even make some new tutorials which we can link here on the landing page which can be something in the Examples section?
- Looks like this just consists of the [Home page](#). At least there doesn't seem to be any links to get anywhere else on this domain.
 - **Top bar**
 - [kubectl usage](#) - a link to the kustomize page in the kubernetes.io docs
 - [Core docs](#) - link to this cli-experimental docs
 - Some anchor links to the rest of the home page
 - **Kubernetes native configuration management**
 - [Install kustomize](#) - old link which I think is the docs folder in the kustomize repo, which redirects to the [current cli-experimental docs](#)
 - [Use with kubectl](#) - links to the introduction page in the current-cli-experimental docs
 - **Overview**
 - Some general kustomize info in short-form which is probably important to reuse somewhere
 - Two links:
 - [apply -k](#) which is broken
 - [standalone](#) which is the github pages site but I've never seen this page before!
 - **Resources**
 - [Kubernetes blog post announcing kustomize](#)
 - [Background whitepaper](#) - this links to the wrong place. I guess it should be [linked here](#) now.
 - [Examples in the kustomize repo](#)
 - [Overview sides](#) - random meetup in Tokyo from 2018. Is this still relevant?

- [Glossary](#) - links to slightly the wrong place but I get redirected. [Current link](#).
- [Use case](#) - wordpress blog from 2018 about “Customizing Upstream Helm Charts with Kustomize”.
- [Kubecon Video](#) - again from 2018 but seems maybe like it'd be more relevant. Not sure if it contains outdated information
- [Kubernetes 1.14](#) - blog post announcing k8s 1.14 which first included kustomize in-built to kubectl. It's a tiny section that talks about this and the blog post links to <https://kubectl.docs.kubernetes.io/>
- [Kustomize introduction](#) - links to the exact same place as the point above! It's the k8s 1.14 announcement blog post again
- **Conference Talks**
 - [Organizing the YAML mess with Kustomize - Florian Assmus](#) - from ContainerDays - July 16th, 2019
 - [Kustomize: Kubernetes Configuration Customization - K8s, Kustomize & Ship SF Meetup](#) - November 17th, 2018
 - [Stop Forking Helm Charts, an Intro to Replicated Ship - K8s, Kustomize & Ship SF Meetup](#) - November 15th, 2018
 - [Kustomize: Deploy Your App with Template Free YAML - Ryan Cox, Lyft](#) - same link as Kubecon Video from the resources section - Dec. 16th, 2018
 - [TGI Kubernetes 072: Kustomize and friends](#) - April 12th, 2019 - 1 hour and 45 minutes long!
- **Projects Integrating Kustomize**
 - Some projects listed here. No idea how up-to-date it is. All the links seem to be correct.
- **Community**
 - [Slack link](#) - references the #kustomize channel
 - [Github link](#)
 - [Official mailing list](#) - seems the link is old but successfully [redirects here](#).

kubernetes-sigs.github.io/kustomize

- Assuming everything we care about from this site [is in here](#), then this seems to be entirely redirects to kubectl.docs.kubernetes.io

Sources of New Content

- [Customizing Kustomize with Client-Side Custom Resources - Katrina Verey, Apple & Jeff Regan, Google](#)
 - [This slide](#) is very nice and might be good to have in the introduction page.
 - [This image](#) is also very nice for conveying what kustomize does in an introduction without spamming lots and lots of yaml code blocks one after the other which may be better for full examples.

- A page where the various patching mechanisms are compared in a table. Which features are part of which approaches? Maybe some associated use-cases mentioned (perhaps linking to full examples)