

Your Project's Website is BAD

But it Doesn't Have to Be :)

Ethan Rose, Apache Ozone PMC Member

**This is Apache Ozone's Current
Website**



Apache Ozone is a highly scalable, distributed storage for Analytics, Big data and Cloud Native applications. Ozone supports S3 compatible object APIs as well as a Hadoop Compatible File System implementation. It is optimized for both efficient object store and file system operations.

It is built on a highly available, replicated block storage layer called Hadoop Distributed Data Store (HDDS).

Applications using frameworks like Apache Spark, YARN and Hive work natively without any modifications.

Ozone is now Generally Available(GA) with 1.4.0 release.

 **SCALABLE**

Ozone is designed to scale to tens of billions of files and blocks and, in the future, even more.

 **SECURE**

Ozone integrates with kerberos infrastructure for access control and supports TDE and on-wire encryption.

 **CONSISTENT**

Ozone is a strongly consistent distributed storage. This consistency is achieved by using protocols like RAFT.

 **MULTI-PROTOCOL SUPPORT**

Ozone supports different protocols like S3 and Hadoop File System APIs.

 **CLOUD-NATIVE**

Ozone is designed to work well in containerized environments like YARN and Kubernetes.

 **HIGHLY AVAILABLE**

Ozone is a fully replicated system that is designed to survive multiple failures.

- Overview
- Getting Started
- Architecture
- Overview
- Ozone Manager
- Storage Container Manager
- Containers
- Datanodes
- Recon
- Features
- Decommissioning
- OM High Availability
- Ozone Erasure Coding
- Ozone Snapshot
- SCM High Availability
- Streaming Write Pipeline
- Merger Container RocksDB in DN
- Prefix based File System Optimization
- Topology awareness
- Quota in Ozone
- Recon Server
- Observability
- Non-Rolling Upgrades and Downgrades
- S3 Multi-Tenancy
- Setup
- Tenant commands
- Access Control

Prefix based File System Optimization

The prefix-based File System Optimization feature supports atomic rename and delete of any directory at any level in the namespace in deterministic/constant time.

This feature can be enabled for each specific bucket that requires it by setting the `--layout` flag to `FILE_SYSTEM_OPTIMIZED` at the time of bucket creation.

```
ozone sh bucket create /<volume-name>/<bucket-name> --layout FILE_SYSTEM_OPTIMIZED
```

Note: File System Optimization favors Hadoop Compatible File System instead of S3 compatibility. Some irregular S3 key names may be rejected or normalized.

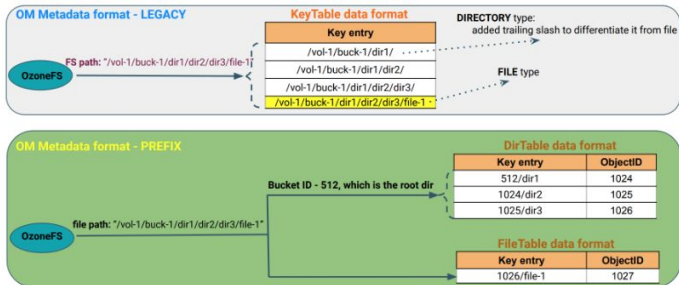
This feature is strongly recommended to be turned ON for Ozone buckets mainly used via Hadoop compatible interfaces, especially with high number of files in deep directory hierarchy.

OzoneManager Metadata layout format

OzoneManager supports two metadata bucket layout formats - Object Store (OBS) and File System Optimized (FSO).

Object Store (OBS) is the existing OM metadata format, which stores key entry with full path name. In File System Optimized (FSO) buckets, OM metadata format stores intermediate directories into `DirectoryTable` and files into `FileTable` as shown in the below picture. The key to the table is the name of a directory or a file prefixed by the unique identifier of its parent directory, `<parent-unique-id>/<filename>`.

KeyTable → DirTable & FileTable

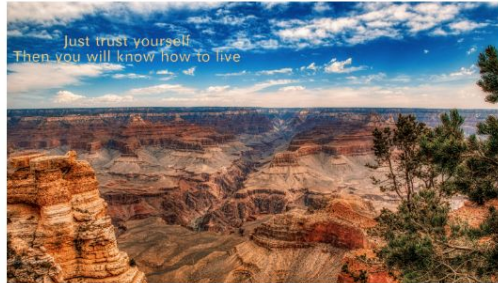


Directory delete operation with prefix layout:

Following picture describes the OM metadata changes while performing a delete operation on a directory.



Release 1.3.0 available



[Download tar.gz](#)

[\(checksum signature\)](#)

[Download src](#)

[\(checksum signature\)](#)

[Documentation](#)

2022 Dec 18

Apache Ozone 1.3.0 is released with following features:

- Erasure Coding - V1 support (Online reconstruction support)
- Container balancer
- Limit number of rocksdb instances to 1 per disk on high-density datanodes
- Ozone S3 Multi-tenancy Support
- S3 Grpc improvements for metadata path
- Filesystem Optimized and Object Store bucket layout types

This is a generally available (GA) release. It represents a point of API stability and quality that we consider production-ready.

Image credit: [Grand Canyon - South Rim - Mather Point](#) by G. Lamar, CC BY 2.0 / Text added to original



SPACE SHORTCUTS

- Ozone Wiki Home
- Try out Ozone
- Contributing to Ozone

PAGE TREE

- Ozone Roadmap
- Ozone Twitter Account
- Project Governance
- Blog posts and talks
- Try out Ozone
- Contributing to Ozone
- Ozone Online Meetup(Mandarin)
- Who Uses Apache Ozone ?

Pages

Ozone Wiki Home

Created by ASF Infrabot, last modified by Ritesh Shukla on May 02, 2023

Ozone is a distributed key-value store that can efficiently manage both small and large files alike. While HDFS provides POSIX-like semantics, Ozone looks and behaves like an Object Store. This series of articles will help you get started contributing to the Apache Hadoop Ozone project.

What is Ozone?

Ozone is a distributed key-value store designed to scale to billions of objects and run on clusters of thousands of nodes.

Ozone supports RPC and S3 APIs as well as a [Hadoop Compatible File System](#) called OzoneFS.

Ozone is a work in progress and currently in alpha state. To try it out, please download a release from [Ozone website](#).

Try it out!

The following page describes your options: [Try out Ozone](#)

Contribute

Ozone is a part of the Apache Hadoop project. The bug tracking system for Ozone is under the [Apache Jira project named HDDS](#).

If you are familiar with contributing to Apache Hadoop, then you already know everything you need to know to start filing Ozone bugs and submitting patches.

If you have never contributed to Apache Hadoop before, then you may find it useful to read [How To Contribute](#).

What can I contribute?

We welcome contributions of:

- Code.** File a bug and submit a patch, or pick up any one of the unassigned Jiras.
 - Newbie Ozone Jiras** - <https://s.apache.org/OzoneNewbieJiras>
 - All open and unassigned Ozone jiras - <https://s.apache.org/OzoneUnassignedJiras>.
- Documentation Improvements.** You can submit improvements to either:
 - Ozone website.** Instructions are here: [Modifying the Ozone Website](#)
 - Developer docs.** These are markdown files checked into the Apache Hadoop Source tree. E.g. <https://github.com/apache/ozone/blob/trunk/hadoop-hdds/docs/content/GettingStarted.md>
 - This wiki page. Please contact us at dev@ozone.apache.org and we can provide you write access to the wiki.
- Unit Tests (JUnit)**
- Acceptance Tests.** The Ozone source code includes a set of docker-based acceptance tests. See this guide for running the available acceptance tests - [\(moved\) Running Ozone Smoke Tests and Unit Tests](#).
- or just file bugs.** Bug reports pointing out broken functionality, docs, or suggestions for improvements are always welcome!

What's Wrong?

A bad website is bad for community adoption

- Appearance
 - People DO judge a book by its cover
 - They expect Ozone to run about as well as the website looks
- Content
 - Docs are disorganized and lacking
 - No continuity between Ozone docs and Wiki

How Did We Get Here?

The website is not run like the project

- Docs layout does not scale with new content
- Updates are ad-hoc, not part of project planning
- No CI
- Poor choice of frameworks

What Can We Do About It?

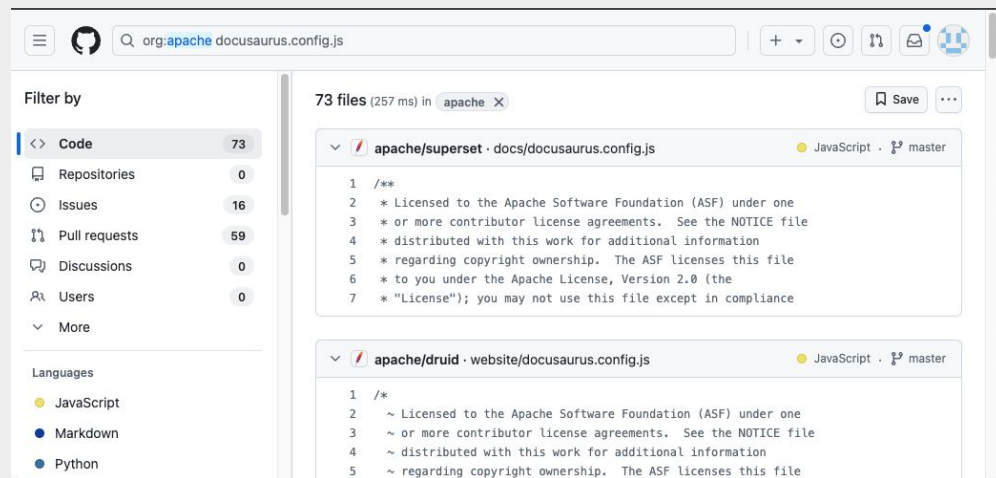
Run the website like the project!

Gathering Requirements

- Tools that look and function well out of the box
 - Most Ozone devs are not web devs
- Extensible Layout
 - Keep docs organized now and in the future
 - Provide a home for stray Wiki content
- Minimal contribution friction
 - Only markdown required
 - Re-use familiar tools

Gathering Inspiration

- ASF has hundreds of projects, each with an open source website and compatibly licensed code!
- Choose a few reference sites with aspects you like:
 - Homepage
 - Search
 - Navigation
 - Docs Layout



The screenshot shows a GitHub search interface. The search bar contains the query 'org:apache docusaurus.config.js'. The results are filtered by 'Code' (73 items). Two results are visible:

- apache/superset · docs/docusaurus.config.js** (JavaScript · master):

```
1 /**
2  * Licensed to the Apache Software Foundation (ASF) under one
3  * or more contributor license agreements. See the NOTICE file
4  * distributed with this work for additional information
5  * regarding copyright ownership. The ASF licenses this file
6  * to you under the Apache License, Version 2.0 (the
7  * "License"); you may not use this file except in compliance
```
- apache/druid · website/docusaurus.config.js** (JavaScript · master):

```
1 /*
2  ~ Licensed to the Apache Software Foundation (ASF) under one
3  ~ or more contributor license agreements. See the NOTICE file
4  ~ distributed with this work for additional information
5  ~ regarding copyright ownership. The ASF licenses this file
```

HDDS-9225: Apache Ozone Website v2

A complete overhaul of Ozone's current website

1. Create design doc outlining the proposal and exit criteria
2. Send to dev@ mailing list for community feedback and engagement
3. Divide into Jira subtasks
4. Create feature branch from ozone-site repo
5. Get to work!

1. Framework

Where to put the code?

- Current Ozone website uses split model:
 - Docs are with Ozone code
 - Rest of website is in ozone-site
 - Confluence Ozone Wiki is completely separate
- **Goal:** Unify website source for a unified website appearance
 - Everything goes in ozone-site repo

1. Framework

Static Site Generator: Docusaurus

- MIT licensed, maintained by Meta open source
- Good docs on how to use it
- Markdown, extensible with JS
- Used by many ASF projects
- Nice default appearance and functionality out of the box
- Builds with standard Node tools



1. Framework

Package Manager: PNPM

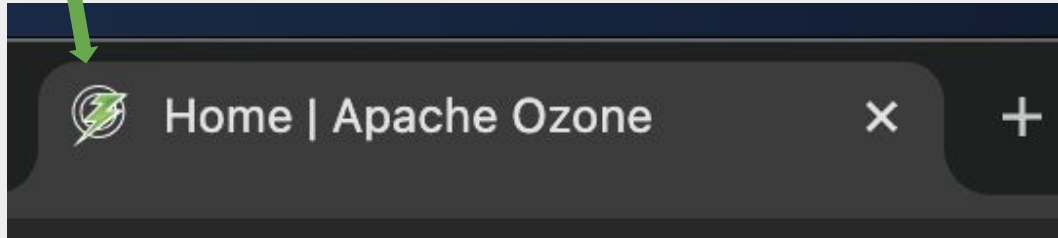
- Already used in Recon: Ozone's observability server
- Chosen over similar alternatives like NPM and Yarn



1. Framework

Favicon

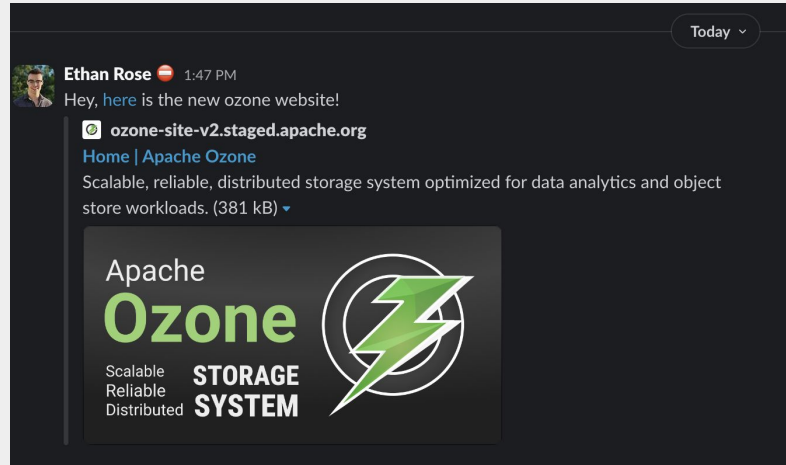
THIS little logo is so simple but so complicated...



1. Framework

Social Card

- Used for webpage previews when links are shared on social media or messaging apps
- A little graphic design goes a long way



1. Framework

Analytics: Matomo

- Supported for Apache projects
- Results are published to analytics.apache.org
- See where and what users are accessing



1. Framework

Search: Algolia

- Indexes your website for users to search
- Supported by Docusaurus with a plugin
- Used by many ASF projects



2. Docs Outline

Who actually reads docs???

Personas: Everyone reading the docs is some combination of:

- User: Interacts with Ozone through a client and API
- Administrator: Interacts with Ozone servers directly
- Developer: Interacts with Ozone by building, modifying, running

A reader should be able to easily avoid docs which do not pertain to them

BUT

We should not duplicate documentation among sections

2. Docs Outline

Don't scare people (yet)

Least technical

Most technical



Version: **Next**

- Overview
- Quick Start >
- Core Concepts >
- User Guide >
- Administrator Guide >
- Troubleshooting >
- System Internals >
- Developer Guide >

2. Docs Outline

Overview

Everyone: What are we looking at?



Version: **Next**

Overview

Quick Start >

Core Concepts >

User Guide >

Administrator Guide >

Troubleshooting >

System Internals >

Developer Guide >

2. Docs Outline

Quick Start

Everyone: Docs are boring, let me try it 

Version: **Next**

Overview

Quick Start >

Core Concepts >

User Guide >

Administrator Guide >

Troubleshooting >

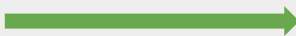
System Internals >

Developer Guide >

2. Docs Outline

Core Concepts

Everyone: Ozone basics for all



Version: **Next**

Overview

Quick Start



Core Concepts



User Guide



Administrator Guide



Troubleshooting



System Internals



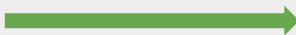
Developer Guide



2. Docs Outline

User Guide

Users: I use Ozone clients or run workloads against Ozone

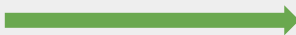


Version: Next	
Overview	
Quick Start	>
Core Concepts	>
User Guide	>
Administrator Guide	>
Troubleshooting	>
System Internals	>
Developer Guide	>

2. Docs Outline

Administrator Guide

Admins: I manage Ozone cluster(s) for users



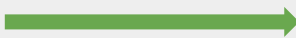
Version: **Next**

- Overview
- Quick Start >
- Core Concepts >
- User Guide >
- Administrator Guide >**
- Troubleshooting >
- System Internals >
- Developer Guide >

2. Docs Outline

Troubleshooting

Everyone: SOS click here!

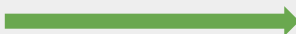


Version: Next	
Overview	
Quick Start	>
Core Concepts	>
User Guide	>
Administrator Guide	>
Troubleshooting	>
System Internals	>
Developer Guide	>

2. Docs Outline

System Internals

Everyone: I'm curious about how Ozone works, beyond what is necessary to run it



Version: Next	
Overview	
Quick Start	>
Core Concepts	>
User Guide	>
Administrator Guide	>
Troubleshooting	>
System Internals	>
Developer Guide	>

2. Docs Outline

Developer Guide

Version: **Next**

Overview

Quick Start >

Core Concepts >

User Guide >

Administrator Guide >

Troubleshooting >

System Internals >

Developer Guide >

Developers: I want to contribute
to Ozone



2. Docs Outline

Staging website doubles as a task list

The screenshot shows a web page with a left sidebar and a main content area. The sidebar contains a navigation menu with items: Overview, Quick Start, Core Concepts, Architecture, Replication, Namespace, Overview (highlighted), Volumes, Buckets, and Security. The main content area has a breadcrumb trail: Home > Core Concepts > Namespace > Overview. The main heading is 'Overview of Ozone's Namespace'. Below the heading is a 'TODO' section with the text: 'File a subtask under HDDS-9857 and complete this page or section.' There is an 'Edit this page' link with a pencil icon. At the bottom, there are two boxes for navigation: 'Previous << Namespace' and 'Next Volumes >>'.

Version: [Next](#)

- Overview
- Quick Start >
- Core Concepts >
- Architecture
- Replication >
- Namespace >
- Overview**
- Volumes >
- Buckets >
- Security >

🏠 > Core Concepts > Namespace > **Overview**

Overview of Ozone's Namespace

TODO: File a subtask under [HDDS-9857](#) and complete this page or section.

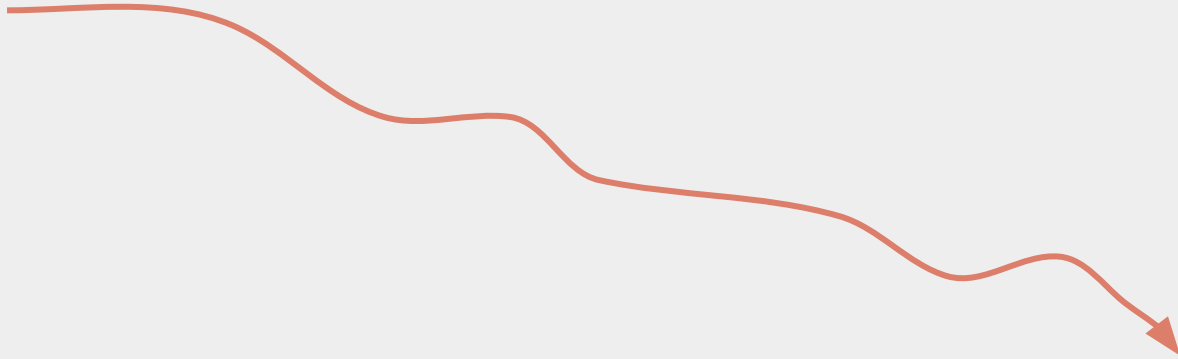
[✎ Edit this page](#)

Previous
[« Namespace](#)

Next
[Volumes »](#)

3. Contributor Experience

Getting people to write docs is already hard.
Don't make it harder



3. Contributor Experience

README.md and CONTRIBUTING.md

- Docs about docs!
- Include a “Quick Start” at the top for minimal friction
- Add further documentation of website details and best practices farther down in CONTRIBUTING.md

Apache Ozone Website Contribution Guide

Apache Ozone is a top-level Apache project and is licensed under the [Apache 2.0 License](#). The bug tracking system for Ozone and its website is under the [Apache Jira project named HDDS](#).

This document summarizes the contribution process.

Quick Start

1. Review the main [Ozone contributing guide](#). Contributing to the new website follows a similar process with a few differences:

3. Contributor Experience

One line preview command

Just git clone and...

```
5s > ~/Repos/apache/ozone-site
docker compose up
[+] Building 0.0s (0/0)
[+] Running 1/0
✓ Container ozone-site-site-1 Created
Attaching to ozone-site-site-1
ozone-site-site-1 |
ozone-site-site-1 | > doc-site@0.1.0 start /ozone-site
ozone-site-site-1 | > docusaurus start --port 3001 "--host" "0.0.0.0"
ozone-site-site-1 |
ozone-site-site-1 | -----
ozone-site-site-1 | Update available 3.3.2 → 3.5.2
ozone-site-site-1 | the latest version, run the following command:
ozone-site-site-1 | @docusaurus/plugin-pwa@latest @docusaurus/module-type-aliases@latest`
ozone-site-site-1 | -----
ozone-site-site-1 | [INFO] Starting the development server...
ozone-site-site-1 | [SUCCESS] Docusaurus website is running at: http://localhost:3001/
ozone-site-site-1 | i Compiling Client
ozone-site-site-1 | ✓ Client: Compiled successfully in 3.96s
ozone-site-site-1 | client (webpack 5.91.0) compiled successfully
```

Open in
browser for
live preview

3. Contributor Experience

Let CI do the nitpicking

- Painful code reviews will turn away contributors and reviewers
- Automate:
 - Pull request title format
 - Jira linking
 - License header checks
 - Spelling
 - Formatting

4. CI

Of course your project has it, why not its website?

- Use the same CI platform as your project
- Reuse checks and templates from your project:
 - PR title check
 - PR template
 - License check
- Make sure error messages are helpful to new contributors
 - CI output can link to relevant parts of CONTRIBUTING.md for help resolving issues.

4.1. CI: Static Checks

License headers: Apache Skywalking-Eyes

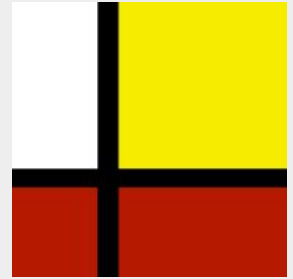
- License checks for any file format without extra dependencies
- Features:
 - Maintained by the Apache Skywalking project
 - Comes with a plug and play GitHub action
 - Flexible configuration



4.1. CI: Static Checks

Spelling: cspell

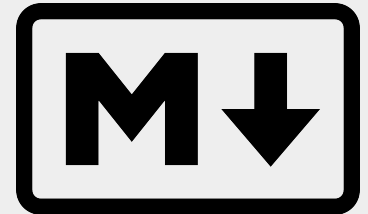
- Never get a “Fix Typo In...” pull request again!
- Features:
 - Import existing dictionaries of jargon
 - Configurable dictionary to add your project’s custom jargon
 - Inline ignores when you need even more jargon



4.1. CI: Static Checks

Markdown formatting: markdownlint

- Make all pages use the same markdown style
- Features:
 - Tons of configurable rules to lock in your markdown format
 - Catch rendering gotchas you didn't know existed
 - Enforce proper capitalization of words
 - It's “Datanode”, not “datanode” or “DataNode”
 - Optionally ignored in `markdown code blocks`



4.2. CI: Build

Environment: Docker + PNPM

- Docker image with all dependencies built in is cached in GitHub actions
- pnpm build runs in a container from this image
- Resulting website is saved as an artifact for the next stage



4.2. CI: Build

Build Hooks: Docusaurus and friends

- Docusaurus supports plugging into the build at various stages
- Use cases:
 - Enforce restrictions on front matter keys with json schema
 - Check format of all URLs when generating the sitemap



4.2. CI: Run

Liveness: curl

- Assume Docusaurus is handling UI functionality correctly
- Just make sure the site can be run and the homepage returns 200

4.3. CI: Publish

How?

- **.asf.yaml**: Branch specific configuration file for ASF related tasks
 - Jira auto linking
 - Repo metadata
 - Website publishing
- ASF supports staging sites:
 - Official site: ozone.apache.org
 - New WIP website: ozone-site-v2.staged.apache.org

4.3. CI: Publish

Steps

1. GitHub action runs on each commit to the website branch
2. The action pushes a commit to the build branch
3. The build branch's `.asf.yaml` configuration indicates the build should be published

5. The Hard Stuff

Homepage

- Docusaurus does not provide a template for the homepage
- Requires web dev and graphic design knowledge not present in the Ozone community
- Ideas to help:
 - Borrowing from other Apache websites
 - Creating a mock-up before trying to write code

5. The Hard Stuff

Translations

- Ozone community maintains Mandarin translations for the documentation
- Docusaurus supports multiple translations, but we must maintain them
- How to track what is missing or stale?

🌐 English ▼



🔍 Search



5. The Hard Stuff

Actually writing docs...

- Docs skeleton contains 200+ pages of content that must be filled in
- Devs are busy and progress is slow
- Ideas to help:
 - Lead by example
 - Distribute the load
 - Ask SMEs directly
 - Ask that all future docs contributions go to new and current website

The End

Go fix your website

<https://github.com/errose28>

erose@apache.org

<https://issues.apache.org/jira/browse/HDDS-9225>

<https://github.com/apache/ozone-site/tree/HDDS-9225-website-v2>