

CHAD CARLSON



Technical education,
documentation, and
engineering leader

Contact

+1 949 929 5143

St. Augustine, FL USA

chadwcarlson@gmail.com

linkedin.com/in/chadwcarlson

github.com/chadwcarlson

Skills

Developer education &
enablement. Technical writing.
Demos and example apps
Technical blogs. Video production.
Docs-as-code. Developer course
creation, training & certification.
Content strategy. Product
feedback. Project management.
Technical support. User
experience & design.

Technical proficiencies

Astro, Mintlify, Hugo, Gitbook,
Docusaurus, Nextra, Redocly,
OpenAPI, Postman, LaTeX, Git.
Django, FastAPI, Wagtail. Gatsby,
Next.js, Express, Strapi. Drupal,
WordPress. Discourse, Forem.

Experienced developer education, documentation, and engineering leader. Brings technical writing, software and web development, DevRel, docs-as-code, as well as managing technical writers and engineers skills in order to deliver high-quality developer enablement materials and software.



Education

Florida Atlantic University Complex Systems & Brain Sciences, 2014-2019

UC Santa Barbara BS, Biopsychology, 2007-2012

Career summary

Platform.sh

Senior Manager, Developer Relations, May 2021 - Feb 2025

- Managed review, information architecture, analytics, feature releases, guides, user product feedback, updates and major learning paths of [Platform.sh](#) and [Upsun](#) public documentation sites. Enforced style guide & standards with testing suites. **Lead contributor. Used:** Hugo, Cypress, Vale, Tailwind, Asciinema, Meilisearch, MariaDB, GA4, Heap, GitHub Actions, JS, Python, Platform.sh.
- Launched the [Upsun Developer Center](#): centralized developer docs, examples, training, technical blog & product feedback strategy. **Used:** Hugo, Vale, Tailwind, Docker, Nix, Fastly, Flexsearch, ExpressJS, goaccess, GitHub Actions, JS, Python, Upsun, Google Analytics, LookerStudio, Heap.
- Delivered the [Upsun Demo Project](#), implementing end-to-end tests through the same core product features asked of users to ensure successful onboarding. **Used:** React, Typescript, Tailwind, Flask, Blackfire player, GitHub actions, Upsun.
- Developed [Upsun Decoupled Symfony Workshop](#) for use in docs following at-event trial run. **Used:** Symfony, Next.js, bash, PostgreSQL, Upsun.
- Managed, maintained, and contributed to 113 [user-facing code examples](#) that made up the primary trial onboarding path, support/solutions examples, and workshop curriculum. Supplemented "official" templates with an external marketplace that integrated community and partner examples into primary catalog for visibility. **Used:** Python, doit, GitHub actions, Platform.sh API, and many individual framework examples in different runtime languages.
- Led support and migration initiative for [Platform.sh Forum](#), from discovery to release. **Used:** Discourse, Python, Forem, Zendesk Community.
- Managed DevRel, engineering, and technical writing direct reports through company acquisition (Blackfire.io), and resulting expanded events, documentation, demo materials, support and content responsibilities.
- Standardized demo examples and documentation product story, leading to more consistent messaging, as well as better process for integrating at-event feedback and tailoring materials for different audience stacks.
- Oversaw technical staffing at events in collaboration with the Events Marketing team, enabling speakers and booth attendees.
- Delivered [technical blog posts, video, and talks](#). Launched/co-hosted the [Change mode podcast](#). **Used:** Hugo, Upsun, Buzzsprout, YouTube, iMovie, Podcastics.

DevRel Engineer, Jan 2019 - May 2021

- Led the migration, retooling, IA, and redesign of the [Platform.sh](#) public docs to Hugo leveraging a custom theme. Acted as primary maintainer of [Platform.sh API](#)

Jira, Asana, ClickUp, Linear, OKRs.
Figma, Canva. Impartner,
Allbound. GitHub actions, GitLab
pipelines. MariaDB, PostgreSQL,
Headless Chrome, Elasticsearch,
Opensearch, Meilisearch. SEO.
Analytics. GA4, LookerStudio.
Metabase. Docker. GitOps. CI/CD.
DevOps. Infrastructure-as-code.

Leadership

Documentation and DevRel
strategy. Remote work and remote
team cadence. Team management
with DevRel, engineering, and
technical writing reports.

References

See LinkedIn recommendations

Interests

Writing, reading, cooking,
neuroscience, evolutionary
biology, film and television

[documentation](#). Revamped onboarding path by delivering Platform.sh Getting Started guides. Brought starter/community example visibility to docs. Improved Python support by expanding central reference, Django, Flask, and Wagtail tutorials, and advanced guides around web servers and dependency management. **Used:** *Hugo, Gitbook, Cypress, Vale, Tailwind, Asciiinema, Redocly, OpenAPI/Swagger, Postman, MariaDB, GitHub Actions, JS, Python, Platform.sh*.

- Authored two developer training paths for the [Developer Certification component of the Platform.sh Agency Partner Program](#). Developed and delivered an additional Strapi-focused workshop. **Used:** *Allbound, iMovie, Audacity, Discourse, Zendesk Community, Strapi, React, GraphQL, PostgreSQL, Platform.sh*.
- Brought in-product visibility to community/partner example code; pushed a general "Deploy on Platform.sh" button. Became trackable [open source code outreach mechanism](#). **Used:** *Python, GitLab pipelines, Marketo, scrapy*.
- Designed demos across major product features, [pulled into main marketing site](#) and at-event iPads for interactive sales/booth enablement. Produced [video recordings](#) of each demo to embed alongside slides and transcripts within the website. **Used:** *iMovie, Audacity, RevealJS, Hugo, Gatsby, Hexnode, Configurator 2*.
- Delivered [technical blog posts](#), [video](#), and [talks](#). Acted as organizer and co-host for the [Deploy Friday podcast](#). **Used:** *Hugo, Platform.sh, Buzzsprout, YouTube, iMovie, Podcastics*.

Florida Atlantic University Graduate Research and Teaching Assistant, 2013-2018

- Led weekly lectures/study groups in psychology and neuroscience courses.
- Presented software models of human motion perception, with conclusions supported by eye-tracking and psychophysics experiments.
- Investigated evolutionary algorithms, reinforcement and other deep learning methods as they relate to the emergence of vision in biological nervous systems.

Work

Major projects

[[View full list & details](#)]

- **Documentation:** Platform.sh/Upsun docs, Upsun Developer Center, marketplace.
- **Curriculum:** Upsun Demo project, workshops, Agency Developer Certification
- **Community:** Community marketplace, Deploy on Platform.sh buttons, The change mode podcast, Deploy Friday podcast, Open source project sponsorship.

Writing

[[View full list & details](#)]

- [Drupal and Upsun: Deploying Drupal CMS and Drupal 11](#)
- [How to Deploy Flutter on Upsun](#)
- [Up\(sun\) and running with Django](#)
- [Surfacing activities with GitHub actions](#)
- [A series of unfortunate commits: software security stories](#)

Talks & demos

[[View full list & details](#)]

- [Build apps, not platforms: Optimized DevOps without platform engineering](#)
- [Single and multi-site search with Meilisearch](#)
- [Beginner Probot: building and hosting GitHub Apps](#)
- [Optimized Django deployments with a Continuous Observability Strategy](#)
- [What you need to know about frontends to decouple](#)

CHAD CARLSON

Senior technical writer,
Developer Documentation

Chad Carlson

St. Augustine, FL USA

chadwcarlson@gmail.com

14 March 2025

As a part of the application process, I'm including samples of my technical writing and documentation work, which you can find below (after my signature).

If you are curious about other forms of documentation I have produced over my career – talks I have given, community education programs I have overseen, demos I've produced, training I've designed – I have done my best to document them on my website:

- [Additional writing samples](#)
- [Talks & produced demos](#)
- [Developer training projects I've managed & contributed to](#)
- [Major documentation and community initiatives I've managed](#)

Sincerely,

Chad Carlson

1. [Drupal and Upsun: Deploying Drupal CMS and Drupal 11](#), *Upsun Developer Center*, devcenter.upsun.com, Jan 2025.

Drupal, the PHP-based Content Management System (CMS) released two major versions in 2024. First, the latest release of Drupal core (Drupal 11), and second, a collection of default and often-used modules that made up a “new direction” for Drupal known as Drupal CMS. A large number of customers deploy Drupal sites on the parent platform-as-a-service offering (Platform.sh), but there had yet to be a comprehensive resource for deploying and working with the latest Drupal on our newest product offering – Upsun. This article sought to describe the process for doing so for both Drupal versions. Both options were tested extensively on both Upsun and Platform.sh to compare behavior/configuration, as well as validate with Drupal experts on our Solutions team.

2. [Python documentation](#), *Upsun public documentation*, docs.upsun.com, Feb 2024

Few people internally at Platform.sh had experience deploying Python web frameworks on our platform, despite so much of the platform itself being written in Python. Because of that, myself and

the DevRel/Docs team as a whole needed to perform most of the investigative work to understand the needs of Python web developers to actually leverage our PaaS solution. Through user interviews remotely and at conferences, we found a wide range of frameworks (Django, FastAPI, Flask, Pyramid) were deployed with an equally larger variety and combination of dependency managers (pip, Pipenv, Poetry) and web servers (Gunicorn, Daphne, Uvicorn, Hypercorn). This left us with a real challenge of how to enable users (perhaps unfamiliar with our platform) to migrate all of these combinations to our unique configuration. Through the workshopping and experimentation lead by myself, this resulted in two notable reference pages that aimed to address all of these potential combinations:

- [Python > Manage dependencies](#)
- [Python > Web servers](#)

Internal Python engineers were invaluable in understanding specifics of running Poetry and asynchronous web servers on our platform in particular.

3. [Build apps, not platforms: Optimized DevOps without platform engineering](#), *Platform.sh YouTube channel*, Jan 2023.

Without a dedicated product marketing team, Developer Relations had the challenge of tailoring a consistent, impactful product value story to developers of potentially any programming language, web framework, industry, and organization maturity level. This resulted in workshopping and redesigning our product story for demos and talks at events, the story supported through our documentation and onboardings, and finally packaged in this final demo surfaced across our resource ecosystem. We came up with the skeleton as a team and through countless workshopped hours with internal stakeholders and real users. I ultimately wrote and produced the final script.

4. [Renaming the default environment](#), *Platform.sh public documentation*, docs.platform.sh, Aug 2023.

At Platform.sh, an increasing number of our customers voiced their desire to have the option to rename the default branches on their projects – those branches which became their production environments on our platform. Like many Git-centric products, `master` as a default name was hardcoded into our system, and even today still requires careful adjustments so one can rename the branch (to `main`, for example) without resulting in data loss on a production site. I wrote this documentation to support this

self-service action, which was helpful in providing the feature request while avoiding destructive, unintended consequences for client data.

5. [Surfacing activities with GitHub actions](#), *Platform.sh blog*, platform.sh/blog, Dec 2021.

Platform.sh, a Platform-as-a-service product, provides integrations to all of the major Git hosting providers in the market. Customers will often set up these integrations, so they can focus development and review within GitLab or GitHub, with certain actions triggering deployment activities on the Platform.sh when relevant. Communicating the status of those activities back into GitHub (in this example), however, was not in 2021 as helpful as it could be. We had just made a larger number of platform activity payloads to the user, but if one of those activities were to fail the integration was not yet at the point to reflect that failure (or its error message) back to GitHub. This article took the reader through writing their own very simple GitHub action to set up such a feedback loop in the interim.

6. [Consequences of restoring backups after the persistent data model has changed](#), *Platform.sh Community Forum*, community.platform.sh, Feb 2020.

An important feature of the Platform.sh platform-as-a-service product is how it handles data. Namely, newly created preview environments inherit production data by default. One positive consequence of this behavior is that backups taken of production environments can be applied to child environments. In this ever-changing space of infrastructure and data, however, it was not always clear of the potentially *negative* consequences of how data worked on the platform. This article sought to demonstrate two important to understand negative consequences of how the Platform.sh data model works, and how if care was not taken to appreciate and prepare regular backups, data loss would be difficult to avoid.