CHAD CARLSON

TECHNICAL WRITING, DEVREL & DOCS LEAD

—

Chad Carlson

St. Augustine, FL USA

chadwcarlson@gmail.com

3 February 2025

Dear Hiring manager,

As a part of my application, you've requested a few writing samples with their context.

I hope the samples below demonstrate my abilities as a technical writer, content engineer, and communicator. Additionally, you can view more examples throughout my writing career on my website.

Sincerely,

Chad Carlson

1. 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.

2. 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.

3. 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.

4. 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.