SKILUP DAYS by: DevOps INSTITUTE

Building cross-site search

Supporting a continuous delivery solution across six sites

Chad Carlson
Technical Writer
Platform.sh
github.com/chadwcarlson
chad.carlson@platform.sh

Agenda

Platform.sh

Search and the great docs migration

Cross-site search on public docs

Public docs + search CD

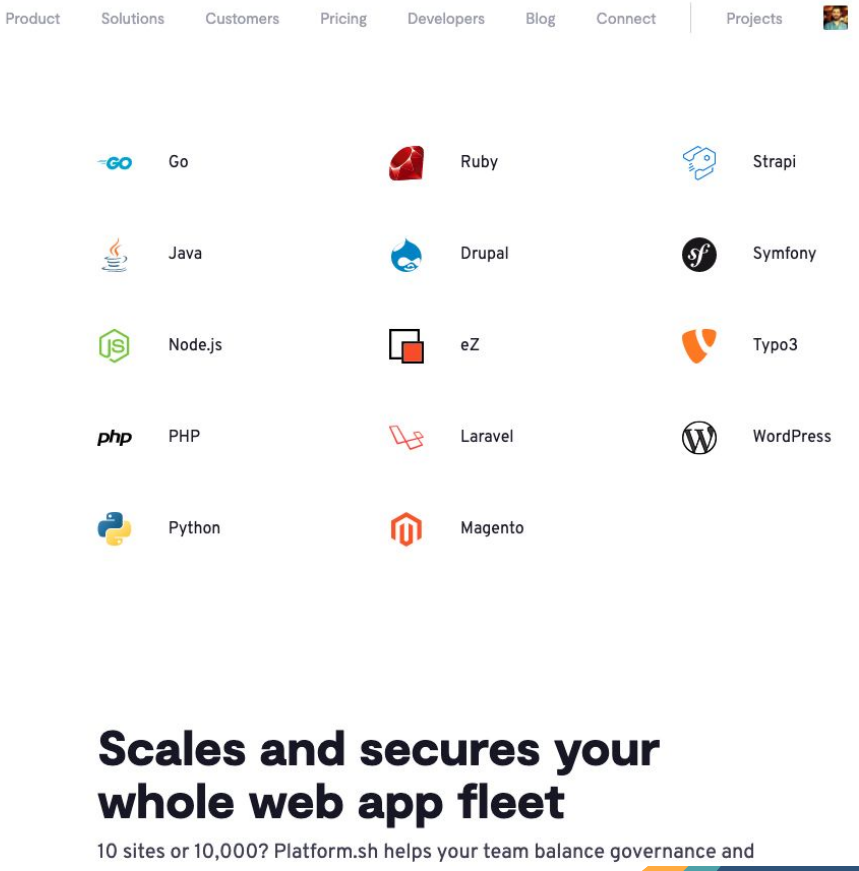
Extending to other sites

Chad Carlson
Technical writer
github.com/chadwcarlson
Writer, programmer, experimenter, ex-scientist.

SKILLup DevOps INSTITUTE DAYS

Platform.sh

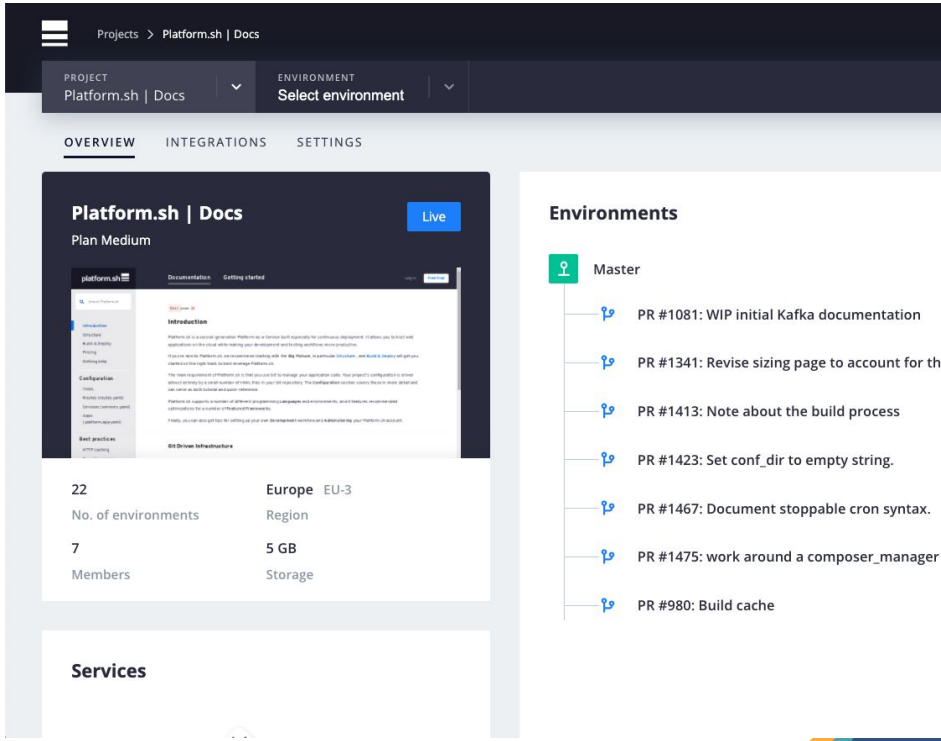
- Platform-as-a-Service = Infrastructure abstraction
  - Polyglot (9 languages)
  - Batteries included services (13)
  - Explicit config for your cluster (3+ YAMLs)
- Built for continuous deployment, tying
  - each branch to an environment
  - each commit to a deployment
  - External integrations to your projects
- Makes merges to production
  - depend on successful builds
  - depend on successful tests
  - depend on successful deployment
- Results in
  - More frequent deployments (experiment)
  - Deploy Friday! (same image)
- Find out more at **docs.platform.sh**



SKILup DAYS

Platform.sh

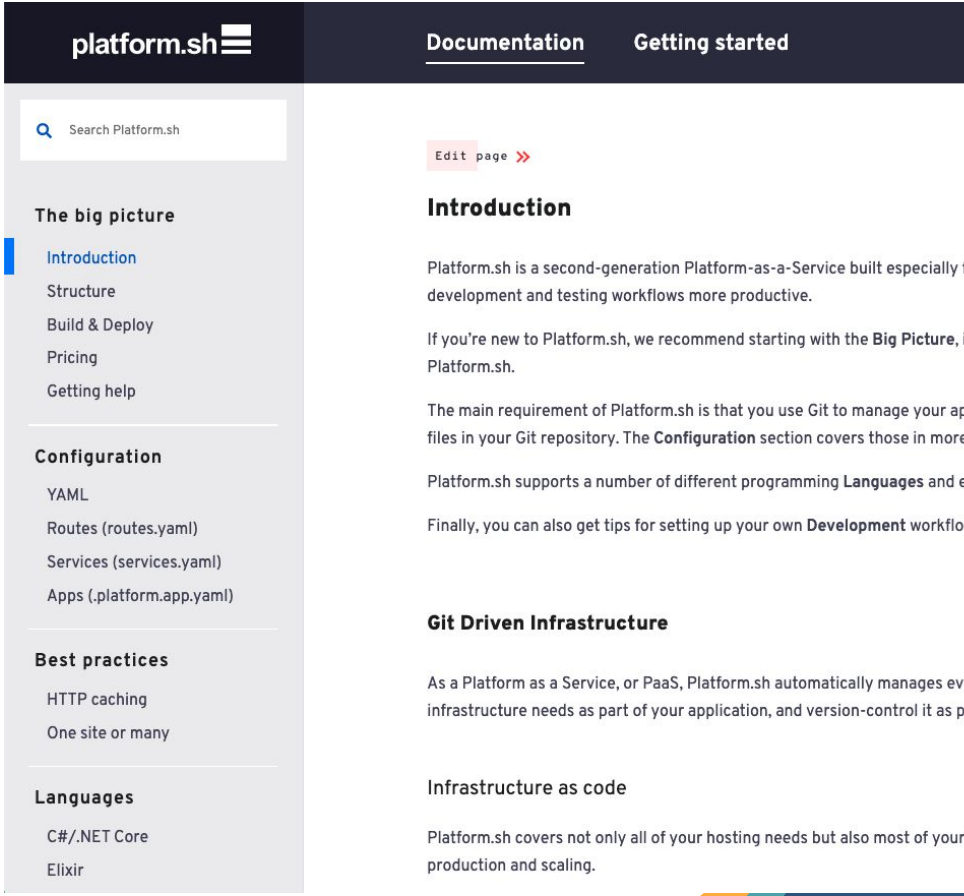
- Platform-as-a-Service = Infrastructure abstraction
  - Polyglot (9 languages)
  - Batteries included services (13)
  - Explicit config for your cluster (3+ YAMLs)
- Built for continuous deployment, tying
  - each branch to an environment
  - each commit to a deployment
  - External integrations to your projects
- Makes merges to production
  - depend on successful builds
  - depend on successful tests
  - depend on successful deployment
- Results in
  - More frequent deployments (experiment)
  - Deploy Friday! (same image)
- Find out more at **docs.platform.sh**



SKILUP DAYS

Docs + search

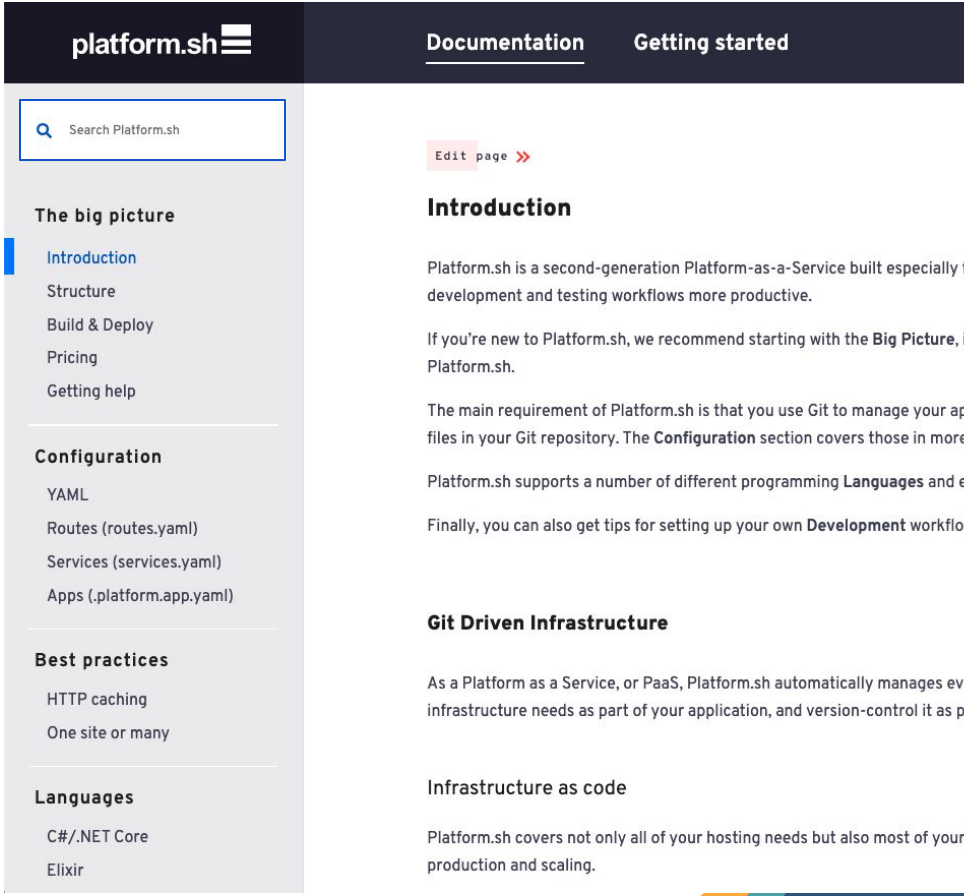
- Previously used Gitbook
  - Node.js
  - Deprecated module ecosystem
  - Used Algolia plug-in for search
- Migrate to Hugo
  - Replicate Algolia?
- Meilisearch (Rust)
  - Executable search engine (multi-app)
  - Need to self-index docs (Hugo custom outputs)
- Side effect: Cross-site search!
  - Could scrape our other sites during build
  - Format documents/pages for Meilisearch
  - Include in the final index



SKILup DAYS

Docs + search

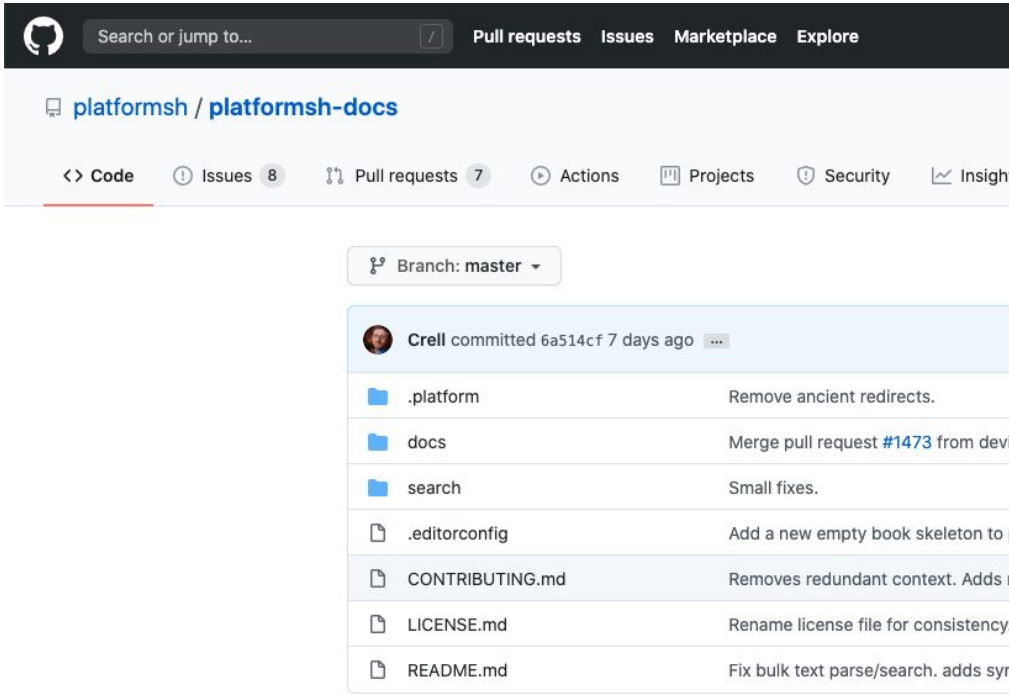
- Previously used Gitbook
  - Node.js
  - Deprecated module ecosystem
  - Used Algolia plug-in for search
- Migrate to Hugo
  - Replicate Algolia?
- Meilisearch (Rust)
  - Executable search engine (multi-app)
  - Need to self-index docs (Hugo custom outputs)
- Side effect: Cross-site search!
  - Could scrape our other sites during build
  - Format documents/pages for Meilisearch
  - Include in the final index



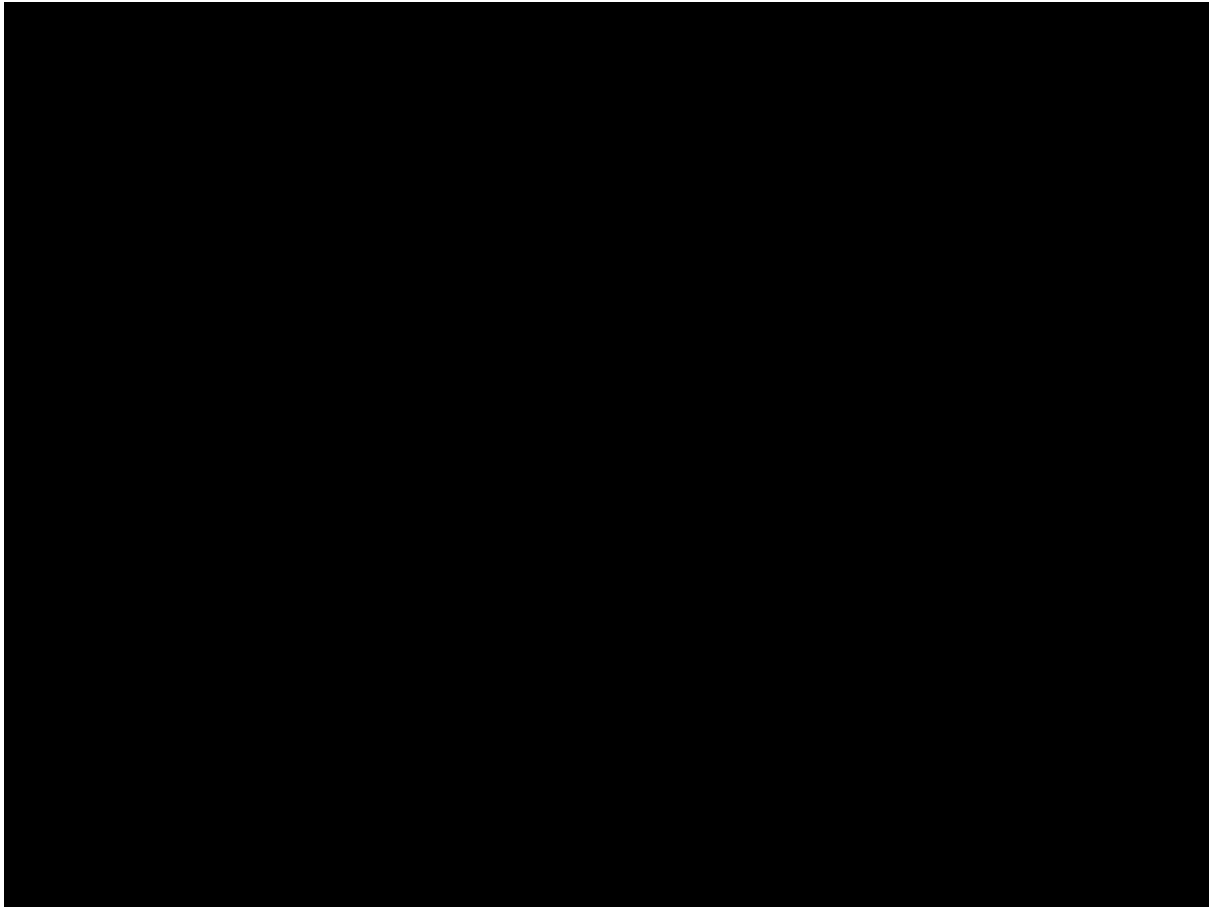
SKILup DAYS

Docs + search

- Previously used Gitbook
  - Node.js
  - Deprecated module ecosystem
  - Used Algolia plug-in for search
- Migrate to Hugo
  - Replicate Algolia?
- Meilisearch (Rust)
  - Executable search engine (multi-app)
  - Need to self-index docs (Hugo custom outputs)
- Side effect: Cross-site search!
  - Could scrape our other sites during build
  - Format documents/pages for Meilisearch
  - Include in the final index



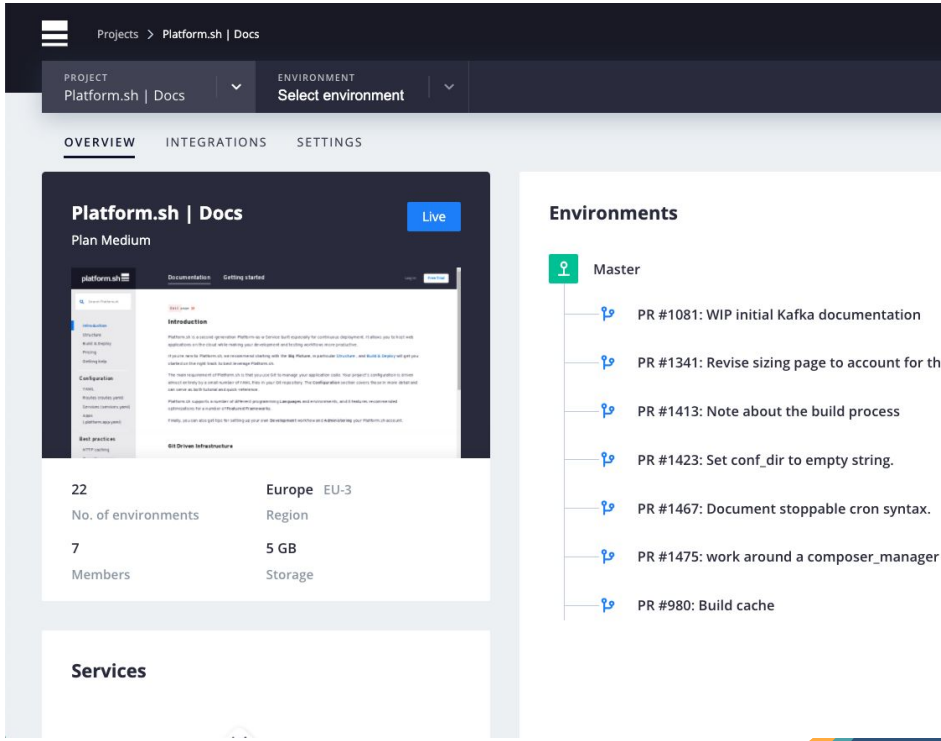
SKILUP DAYS



SKILup DAYS

XSS CD on docs

- Each pull request triggers new environment creation
- Each commit = new deployment
- Limits on merges, dependent on successful deploy
- Builds/deploys themselves depend on:
  - Indexing for search engine
    - If scraper tests fail, deploy fails
    - If scrape fails, deploy fails
    - If self-index fails, deploy fails
  - Posting to Meilisearch
    - Invalid document format = failed deploy
    - Unsuccessful POST = failed deploy
  - Hugo build
    - Fails if index did not create private key
- All of these must pass to be able to merge



SKILup DAYS

XSS CD on docs

- Each pull request triggers new environment creation
- Each commit = new deployment
- Limits on merges, dependent on successful deploy
- Builds/deploys themselves depend on:
  - Indexing for search engine
    - If scraper tests fail, deploy fails
    - If scrape fails, deploy fails
    - If self-index fails, deploy fails
  - Posting to Meilisearch
    - Invalid document format = failed deploy
    - Unsuccessful POST = failed deploy
  - Hugo build
    - Fails if index did not create private key
- All of these must pass to be able to merge

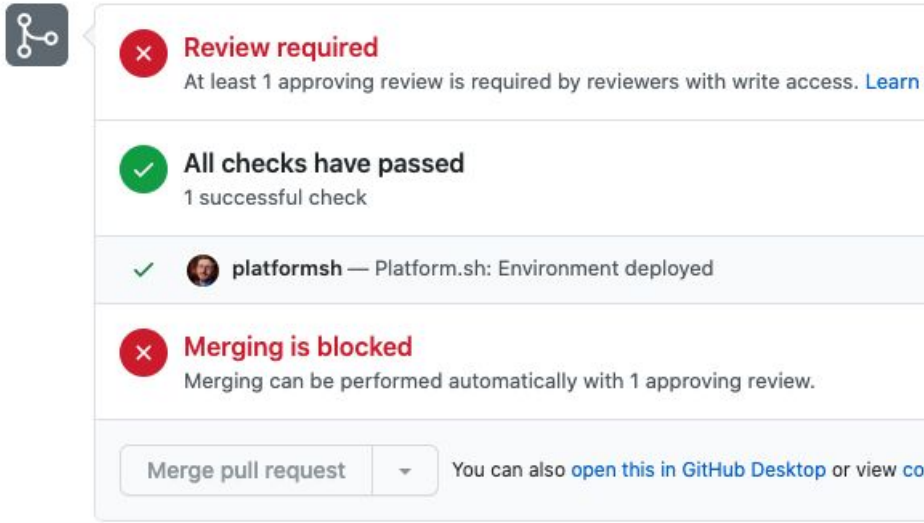
```yaml
name: 'search'

type: 'golang:1.14'

hooks:
    build: !include
        type: string
        path: build.sh
    post_deploy: !include
        type: string
        path: post_deploy.sh

web:
    commands:
      # Run the Meilisearch server
      start: "./meilisearch --http-addr localhost:${PORT}"
```

```bash
## build.sh ##

# Install Meilisearch
curl -L https://install.meilisearch.com | sh

# Build & test the indexer app
go test
go build

###############

## post_deploy.sh ##

# Build the index and post to meilisearch
./pshindex --index-config=config/index.yaml --meili-config=config/meili
```

SKILup DAYS

XSS CD on docs

- Each pull request triggers new environment creation
- Each commit = new deployment
- Limits on merges, dependent on successful deploy
- Builds/deploys themselves depend on:
  - Indexing for search engine
    - If scraper tests fail, deploy fails
    - If scrape fails, deploy fails
    - If self-index fails, deploy fails
  - Posting to Meilisearch
    - Invalid document format = failed deploy
    - Unsuccessful POST = failed deploy
  - Hugo build
    - Fails if index did not create private key
- All of these must pass to be able to merge



SKILUP DAYS

platform.sh

Documentation | Getting started

Search Platform.sh

The big picture
- Introduction
- Structure
- Build & Deploy
- Pricing
- Getting help

Configuration
- YAML
- Routes (routes.yaml)
- Services (services.yaml)
- Apps (.platform.app.yaml)

Best practices
- HTTP caching
- One site or many

Languages
- C#/.NET Core
- Elixir

Edit page

Introduction

Platform.sh is a second-generation Platform-as-a-Service built especially for continuous deployment. It allows you to host web applications on the cloud while making your development and testing workflows more productive.

If you're new to Platform.sh, we recommend starting with the **Big Picture**, in particular Structure , and Build & Deploy will get you started on the right track to best leverage Platform.sh.

The main requirement of Platform.sh is that you use Git to manage your application code. Your project's configuration is driven almost entirely by a small number of YAML files in your Git repository. The **Configuration** section covers those in more detail and can serve as both tutorial and quick-reference.

Platform.sh supports a number of different programming **Languages** and environments, and it features recommended optimizations for a number of **Featured Frameworks**.

Finally, you can also get tips for setting up your own **Development** workflow and **Administering** your Platform.sh account.

Git Driven Infrastructure

As a Platform as a Service, or PaaS, Platform.sh automatically manages everything your application needs in order to run. That means you can, and should, view your infrastructure needs as part of your application, and version-control it as part of your application.

Infrastructure as code

Platform.sh covers not only all of your hosting needs but also most of your DevOps needs. It is a simple, single tool that covers the application life-cycle from development to

platform.sh

Product | Solutions | Customers | Pricing | Developers | Blog | Connect | Projects

Polyglot? Hyperpolyglot? We glot you.

Platform.sh supports an array of cool languages and frameworks. Monoliths. Microservices. So you can keep your options open. Flexible. And future-proof.

Deploy 60+ software packages with a click »

Go, Ruby, Strapi, Java, Drupal, Symfony, Node.js, eZ, Typo3, PHP, Laravel, WordPress, Python, Magento

Search or jump to... | Pull requests | Issues | Marketplace | Explore

Platform.sh Templates

Project Templates maintained by Platform.sh

The Clouds | https://platform.sh/

Repositories 76 | Packages | People 6 | Teams | Projects | Settings

Find a repository... | Type: All | Language: All | Customize pins | New

mattermost
A simple Mattermost template for Platform.sh
Shell 0 0 0 0 Updated 1 hour ago

elastic-apm
Elastic APM template for Platform.sh.
CSS 0 0 0 0 Updated 1 hour ago

hugo
Hugo template for Platform.sh.
golang
HTML 1 0 0

backdrop
Backdrop template for Platform
PHP 0 0 0

Top languages: PHP, Java, Python, JavaScript, HTML

People 6

Invite someone

Platform.sh Playbook

- Welcome to Platform.sh
- Tools used @ Platform.sh
- Training
- Platform Standard (1.x)
- Platform Standard (3.x)
- Platform Dedicated
- Auth
- Support
- Accounts
- Customer Care
- Ops
- Finance
- Marketing
- People Ops
- Sales
- Security
- CLI

Search Platform.sh

Scales and secures your whole web app fleet

10 sites or 10,000? Platform.sh helps your team balance governance and

TEMPLATES

- Types of templates
- Template Builder
- External Templates
- Template Definition files
- Template indexer

Types of templates

Templates are Git repositories that a user may use to initialize a Platform.sh project. They are all intended to be works-out-of-the-box, or as close to that as is feasible. Templates include fully-ready applications (Drupal, Jenkins, WordPress), frameworks (Symfony, Ruby on Rails, Spring Boot), and proof-of-concept templates for specific language configurations. (The latter may be removed at some point in the future.)

Templates are primarily intended as starter kits, not examples, although they can and should be viewed as our officially recommended way to run certain systems.

There is no intention or expectation that users will update their projects post-install from a template. If they wish to do so (to get updated glue code, for instance), that is an entirely manual process.

At this time, there are two sources of templates: Template Builder and External.

Template Builder

platform.sh

Platform.sh Community

Welcome to the Platform.sh Community! Ask or answer any questions in the Q&A section, and browse the How-To Guides to learn how to do cool things with Platform.sh.

Quick Links

- Getting Started
- Going Live
- Troubleshooting
- Support on Slack

Questions & Answers | Latest | Top | Edit | New Topic

All Time

Topic	Replies	Views	Activity
Is it possible to kill a build or deploy? (Questions & Answers)	1	920	Mar '19
Using a custom certificate and LetsEncrypt certificates on the same project? (Questions & Answers; letsencrypt, tls, ssl, certificates)	1	315	Jul '19
What does "WARNING: [pool web] server reached max_children setting (2), consider raising it" mean? (Questions & Answers)	1	961	May '19
How do I fix 'fatal: the remote end hung up unexpectedly' while pushing from a Bitbucket pipeline to Platform.sh?	2	668	Mar 12

platform.sh

Search...

- Domain Management
- Cert Management
- Project Variables
- Repository
- Third-Party Integrations
- Support
  - GET Platform.sh support ticket comment
  - PUT Create a new Platform.sh support ticket comment
  - GET Description is the first comment of a ticket
  - GET A complete list of attachments for a ticket
  - GET Platform.sh support tickets
  - PUT Create a new Platform.sh support ticket
  - GET Return a list of available priorities for a license

Environment

On Platform.sh, an environment encompasses a single instance of your entire application stack, the services used by the application, the application's data storage, and the environment's backups.

In general, an environment represents a single branch or merge request in the Git repository backing a project. It is a virtual cluster of read-only application and service containers with read-write mounts for application and service data.

On Platform.sh, the `master` branch is your production environment–thus, merging changes to master will put those changes to production.

Activate an environment

Set the specified environment's status to active

AUTHORIZATIONS: OAuth2

PATH PARAMETERS

- projectId (required) — string
- environmentId (required) — string

Responses

default

POST /projects/{projectId}/environments/{envi...

Response samples

default

Content type: application/json

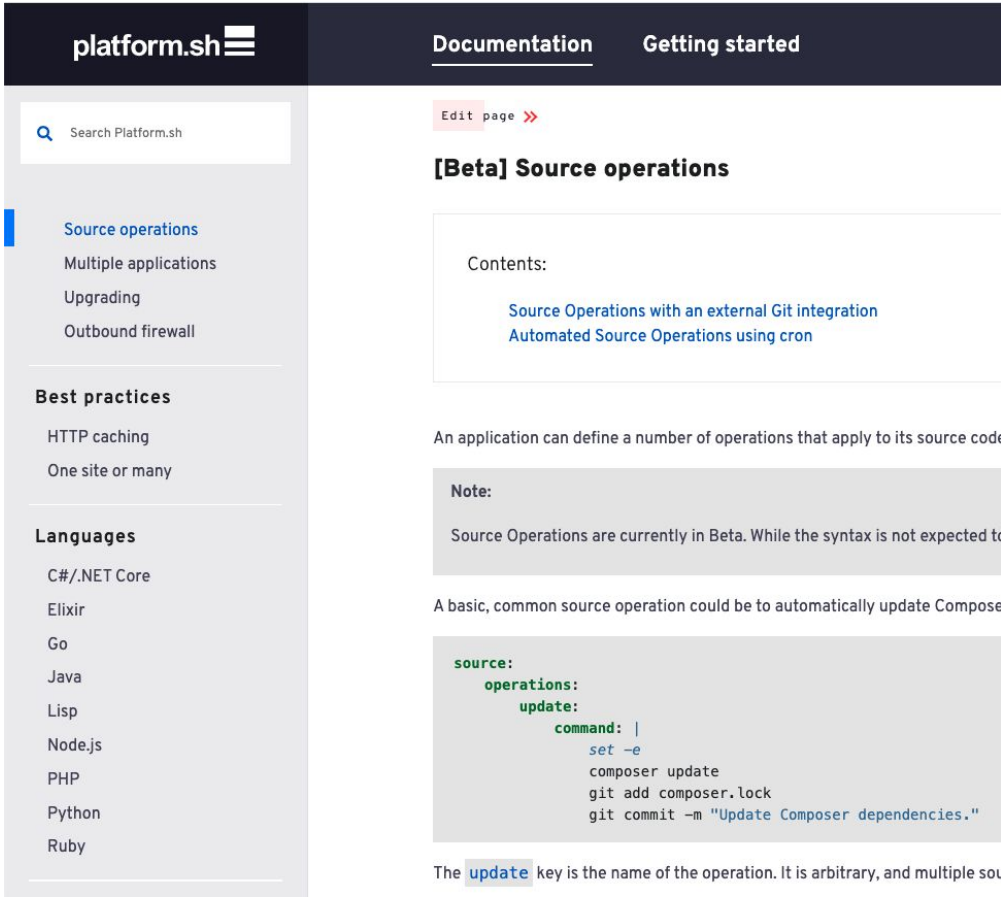
Copy | Expand all | Collapse all

```
{
  "status": "string",
  "code": 0
}
```

SKILUP DAYS

XSS for all!

- Replicate the same model on each site
- Prioritize results for each with independent search server (all become multi-apps)
- We need a way to make index updates easy, regular
  - Source operations
  - Daily cron tests + re-indexes all sites
  - Can be triggered manually via CLI/API
- Now we have
  - A fleet of sites that can synchronize
  - Contain daily updates for each
  - An endpoint created to trigger on all, for big content launches (new language released; EOL announcements, etc.)
  - Each with inbuilt tests where merging to production is blocked if index generation, index POST, or site build fail anywhere.



SKILUP DAYS

XSS for all!

- Replicate the same model on each site
- Prioritize results for each with independent search server (all become multi-apps)
- We need a way to make index updates easy, regular
  - Source operations
  - Daily cron tests + re-indexes all sites
  - Can be triggered manually via CLI/API
- Now we have
  - A fleet of sites that can synchronize
  - Contain daily updates for each
  - An endpoint created to trigger on all, for big content launches (new language released; EOL announcements, etc.)
  - Each with inbuilt tests where merging to production is blocked if index generation, index POST, or site build fail anywhere.

```
# .platform.app.yaml
crons:
    update:
        # Run the 'update' source operation every day at 1:00am.
        spec: '0 1 * * *'
        cmd: |
            set -e
            if [ "$PLATFORM_BRANCH" = update-index ]; then
                platform environment:sync code data --no-wait --yes
                platform source-operation:run update_index --no-wai
            fi
source:
    operations:
        update_index:
            command: |
                set -e
                go test
                ./pshindex --index-config=config/index.yaml --meili
                git add .
                git commit -m "Update Meilisearch index."
```

SKILUP DAYS

THANK YOU!

Meet me in the Network
Chat Lounge for questions

SKILup DAYS