클로저 EDN으로 HTML 이력서 생성하기

디노티시아 김대현
@hatemogi

발표 내용

HTML 이력서를 생성해 공유하는 방법의 하나

최적의 방법 아니고, 그저 하나의 방법

클로저 활용한 방법

RATIONALE

뭘 왜 만들었나?

전개

몇 달 전, 이직을 위해 PDF 이력서를 작성

- PDF 버전 이력서는 "진지한" 지원시 전달

HTML 버전 이력서를 만들자

- "가벼운" 공유 목적

HTML 버전 이력서를 뭘로 만들까?

날것의 HTML? → No, No, No!

GitHub 마크다운? → Good

예전 홈페이지에 썼던 ELM? → Pretty Good

Clojure → Excellent

간결한 매력의 Hiccup이 기억남

클로저 Clojure

현대판 LISP 언어 | LISt Processing

JVM 환경, JS 환경, ...

동적 타입 언어

소스코드를 List 자료구조로 작성

소스코드를 표현하는 EDN (superset)

```
(defn -main [] (println "Hello, World"))
```

resume.hatemogi.com

보유기술

우수한 기술

- Scala
- Clojure
- Haskell
- Elm
- Java

충분한 기술

- TypeScript
- JavaScript
- Kotlin
- PostgreSQL
- Amazon Web Services
- Google Cloud Platform
- Netty
- Docker

경험한 기술

- Rust
- Go
- Python
- C
- C++
- Pascal
- Assembly
- Kubernetes

주관적 자기 평가입니다. 당장 업무 환경에서 주변 다른 개발자들에게 설명하며 일할 수 있는 수준은 **우수한 기술**, 스스로 현업 환경에서 제 몫을 하며 활용할 수 있는 **충분한 기술**, (재)적응이 필요하지만 몇 주 안에 업무 환경에서 홀로 역량을 발휘할 수 있는 수준은 **경험한 기술**로 구분해 적었습니다.

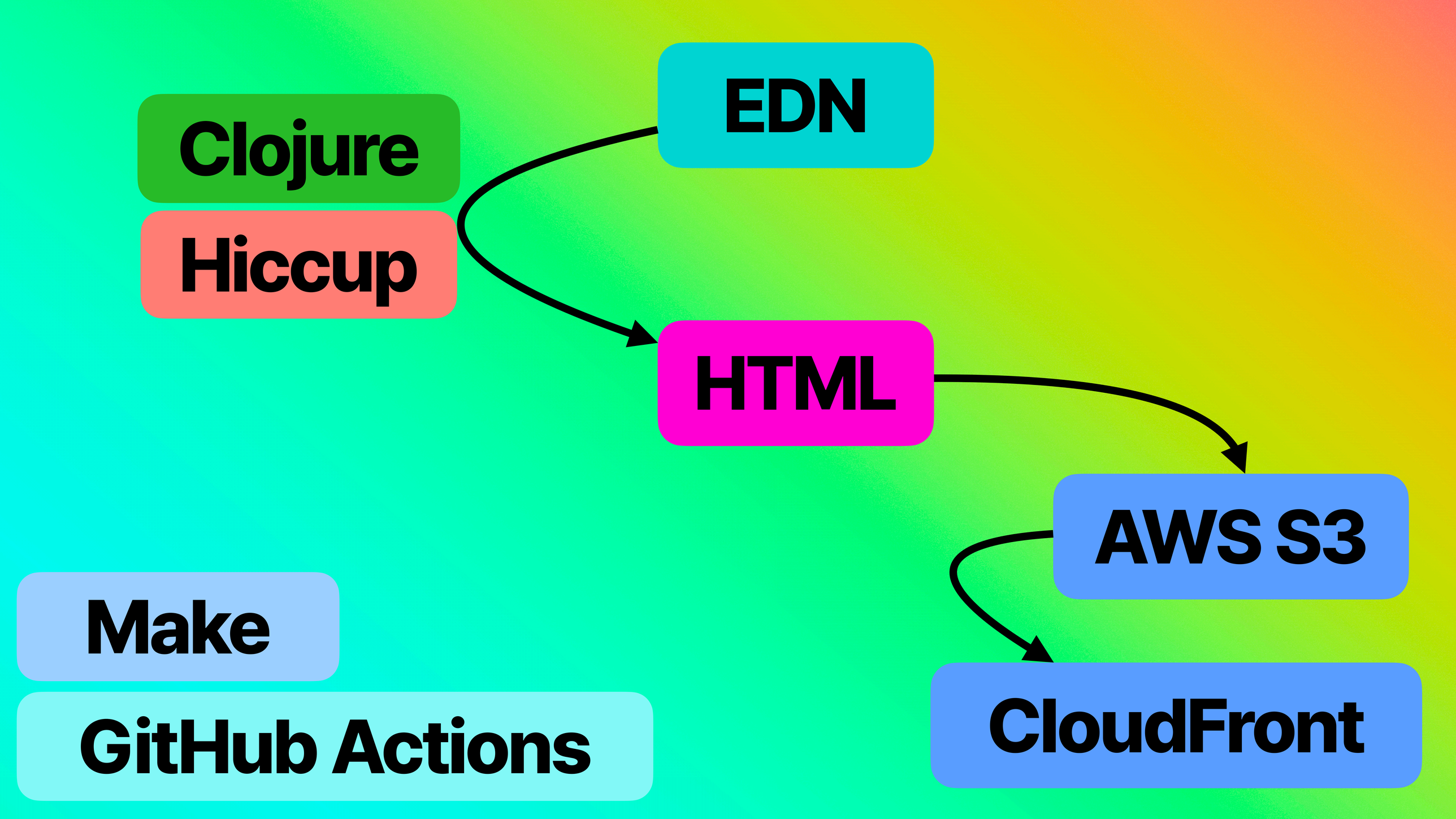
외국어

🇺🇸 English 듣기/읽기 상급, 말하기/쓰기 중급. TOEFL CBT 260

🇯🇵 日本語 듣기/말하기 중급. 읽기/쓰기 초급

발표 경험

인프콘2023 – 함수형 프로그래밍 3대장 경험기: 클로저, 스칼라, 하스켈
LiftIO 2022 – 두 다리 뻗고 자는 (탄탄한) 함수형 프로그래밍
LiftIO 2021 – 연속된 우연으로 꾸려진 개발팀의 함수형 Scala 활용기
2021, 라인 백엔드 개발자의 함수형 프로그래밍 언어 실전 사용기
2017, 제6회 리스프 세미나 — 클로저 소개
PyCon2016 라이트닝톡 — 개발하는 라이더의 연비계산
DevOn2013 — 사내 Git저장소 허브 개발사례


Clojure
Hiccup
EDN
HTML
AWS S3
CloudFront
Make
GitHub Actions

EDN

EXTENSIBLE DATA NOTATION

EDN

Extensible Data Notation

https://github.com/edn-format/edn

A superset of edn is used by Clojure to represent programs

JavaScript / JSON

Clojure / EDN

EDN 기본 요소

nil

booleans | true false

strings | "Hello" "World"

characters | \h \return

symbols | 지시자 . * + ! - _ ? $ % = < > 포함 가능

keywords | :fred

integers | 123456

floating point numbers | 3.141592

EDN 복합 요소

lists | (a b 42)

vectors | [a b 42]

maps | {:a 1, "foo" :bar, [1 2 3] four}

sets | #{a b [1 2 3]}

HICCUP

CLOJURE TO HTML

Hiccup

Clojure 라이브러리

평범한 Clojure 코드(EDN) → HTML 변환

https://github.com/weavejester/hiccup

Hiccup is a library for representing HTML in Clojure. It uses vectors to represent elements, and maps to represent an element's attributes.

[:div]

Vectors that begin with a keyword represent HTML elements

[:div [:span] [:span]]

Any value placed after the keyword is treated as the body of the element. This allows elements to be nested.

```
<div>
  <span></span>
  <span></span>
</div>
```

[:div "안녕하세요"]

Strings are rendered as plain text

안녕하세요

[:div "굿" "이브닝"]

adjacent strings are concatenated together.

굿이브닝

[:div {:class "foo"} "안녕하세요"]

A map can be optionally specified as the second value in the vector. This map represents the attributes of the element.

안녕하세요

[:div 42]

Values of types that have no special meaning are converted into strings before being rendered.

[:div "42"]

[:div '("foo" "bar")]

Seqs (including lists) are automatically expanded out

[:div "foo" "bar"]

[:ul (for [i (range 3)] [:li i])]

Seqs (including lists) are automatically expanded out

```
<ul>
  <li>0</li>
  <li>1</li>
  <li>2</li>
</ul>
```

[:div#foo "안녕하세요"]

The # character can be used to concisely specify the id attribute

안녕하세요

[:div.foo "안녕하세요"]

the . character can be used to specify the class attribute.

안녕하세요

[:div#foo.bar.baz "안녕하세요"]

these two syntaxes can be combined to produce elements with both an id and one or more classes.

안녕하세요

```
(def 취미
  [:section [:h2 "취미"]
   [:article.취미
    [:div.컬럼.고딕
     [:div "달리기"] [:div "누적 2,800km 달림"]
     [:div "하프마라톤 PB 1시간 48분 "
      [:a {:href "img/half-marathon.png" :title "기록증"}
       [:i {:class "fa-solid fa-certificate"}]]]]]])

(def 꼬리말
  [:footer
   [:a {:href "https://github.com/hatemogi/resume.hatemogi.com"}
    [:i {:class "fa-brands fa-github"}]]])

(def 본문 [:main 경력 스킬셋 발표경험 사이드프로젝트 학력 교육 인기글 취미])

(def 바디 [:body 머리말 본문 꼬리말])

(→ (html5 헤더 바디) str print)
```

Clojure CLI

BUILD ONCE

clj -M {파일명 or namespace}

clj -M resume.clj

-M 옵션 | Run a main namespace or script

https://clojure.org/reference/clojure_cli

의존성 관리 deps.edn

deps.edn

```
1   {:deps
2    {hiccup/hiccup {:mvn/version "2.0.0-RC3"}}}
3
```

deps.edn

https://clojure.org/reference/deps_edn

REPL

(loop (print (eval (read))))

VSCode 확장 프로그램 for Clojure

Calva

Start a REPL (jack-in)

VS CODE DEMO

시간이 된다면...

BUILD 반복

MAKEFILE
FSWATCH

Makefile

전통적인 빌드 툴

CLI 커맨드 나열해서 빌드 작업

make build

make watch

make clean

make deploy

fswatch | 파일 변경 감지 CLI 툴

https://emcrisostomo.github.io/fswatch/

기능

관심있는 파일이 변경되면, 해당 파일명을 출력(stdout)해주는 프로그램

활용 목적

개발 중인 파일을 변경할 때마다 변환해서 웹페이지를 다시 읽고 싶다

emcrisostomo.github.io/fswatch/

View on GitHub

fswatch

A cross-platform file change monitor with multiple backends: Apple OS X *File System Events* API, *BSD *kqueue*, Solaris/Illumos *File Events Notification*, Linux *inotify*, Microsoft Windows and a stat()-based backend.

tar.gz .zip

`fswatch` is a file change monitor that receives notifications when the contents of the specified files or directories are modified. `fswatch` implements four kinds of monitors:

- A monitor based on the *File System Events API* of Apple OS X.
- A monitor based on *kqueue*, an event notification interface introduced in FreeBSD 4.1 and supported on most *BSD systems (including OS X).
- A monitor based on *inotify*, a Linux kernel subsystem that reports file system changes to applications.
- A monitor based on *File Events Notification*, a Solaris/Illumos kernel API that reports file events.
- A monitor based on *ReadDirectoryChangesW*, a Microsoft Windows API that reports changes to a directory.
- A monitor which periodically stats the file system, saves file modification times in

Makefile

```
SRC=resume.clj
OUT=docs/index.html
S3_BUCKET=resume.hatemogi.com
CLOUD_FRONT_DISTRIBUTION=EOG2UY17N1S9W

build: ${SRC}
  clj -M ${SRC} | tee ${OUT}
  cp style.css docs/

watch:
  fswatch ${SRC} docs/style.css | xargs -L1 -I{} sh -c "clj -M ${SRC

deploy: build
  aws --profile resume s3 cp docs s3://${S3_BUCKET}/ --recursive --e
  aws --profile resume cloudfront create-invalidation --no-cli-pager
```

DEPLOYMENT

생성한 **HTML** 파일을 아무나 브라우저로 접근해서 볼 수 있게 하자

AWS S3

정적 웹사이트 호스팅 기능

HTML, CSS, JS 파일들 버켓에 올려놓고 HTTP로 서비스 가능

AWS CloudFront 통해 커스텀 도메인으로 HTTPS 서비스 가능

https://resume.hatemogi.com/

AWS CloudFront

CDN 서비스

HTTP/S 보안 연결 처리

커스텀 도메인 지원

TLS 공인인증서 무료 발급 및 자동 갱신

S3에 있는 정적 파일로도 웹서비스 가능

GitHub Actions

YAML파일로 할일을 기술

.github/workflows/main.yml

해당 GitHub 저장소에 지정한 브랜치에 커밋이 push될 때마다

컴파일/테스트/배포 등의 원하는 작업 실행

.github > workflows > main.yml

```yaml
name: Build

on: [push]

env:
  S3_BUCKET: resume.hatemogi.com
  CLOUD_FRONT_DISTRIBUTION: EOG2UY17N1S9W

jobs:
  build-index:
    runs-on: ubuntu-latest

    steps:
      - name: Prepare java
        uses: actions/setup-java@v4.0.0
        with:
          distribution: 'corretto'
          java-version: '17'
```

github.com/hatemogi/resume.hatemogi.com/actions

hatemogi / **resume.hatemogi.com**

Type / to search

Code | Issues | Pull requests | **Actions** | Projects | Security | Insights | Settings

Actions

New workflow

All workflows

Build

Management

- Caches
- Attestations
- Runners

All workflows

Showing runs from all workflows

Filter workflow runs

37 workflow runs Event Status Branch Actor

발표용 예제 추가 **Build** #54: Commit 114631f pushed by hatemogi	main	11 hours ago 3m 43s	...
LLM 수업 수료증 추가 **Build** #53: Commit 0bb2066 pushed by hatemogi	main	last month 2m 49s	...
favicon 추가 **Build** #52: Commit b5485bb pushed by hatemogi	main	2 months ago 35s	...
핵심가치는 아직 정리하지 않음 **Build** #51: Commit 3302129 pushed by hatemogi	main	2 months ago 35s	...
마라톤 PB갱신. 디노티시아 경력 추가 **Build** #50: Commit 5ae8e88 pushed by hatemogi	main	2 months ago 3m 12s	...

github.com/hatemogi/resume.hatemogi.com/actions/runs/10059078

hatemogi / **resume.hatemogi.com**

Type / to search

Code | Issues | Pull requests | **Actions** | Projects | Security | Insights | Settings

← Build

발표용 예제 추가 #54

Re-run all jobs

Summary

Jobs

build-index

Run details

Usage

Workflow file

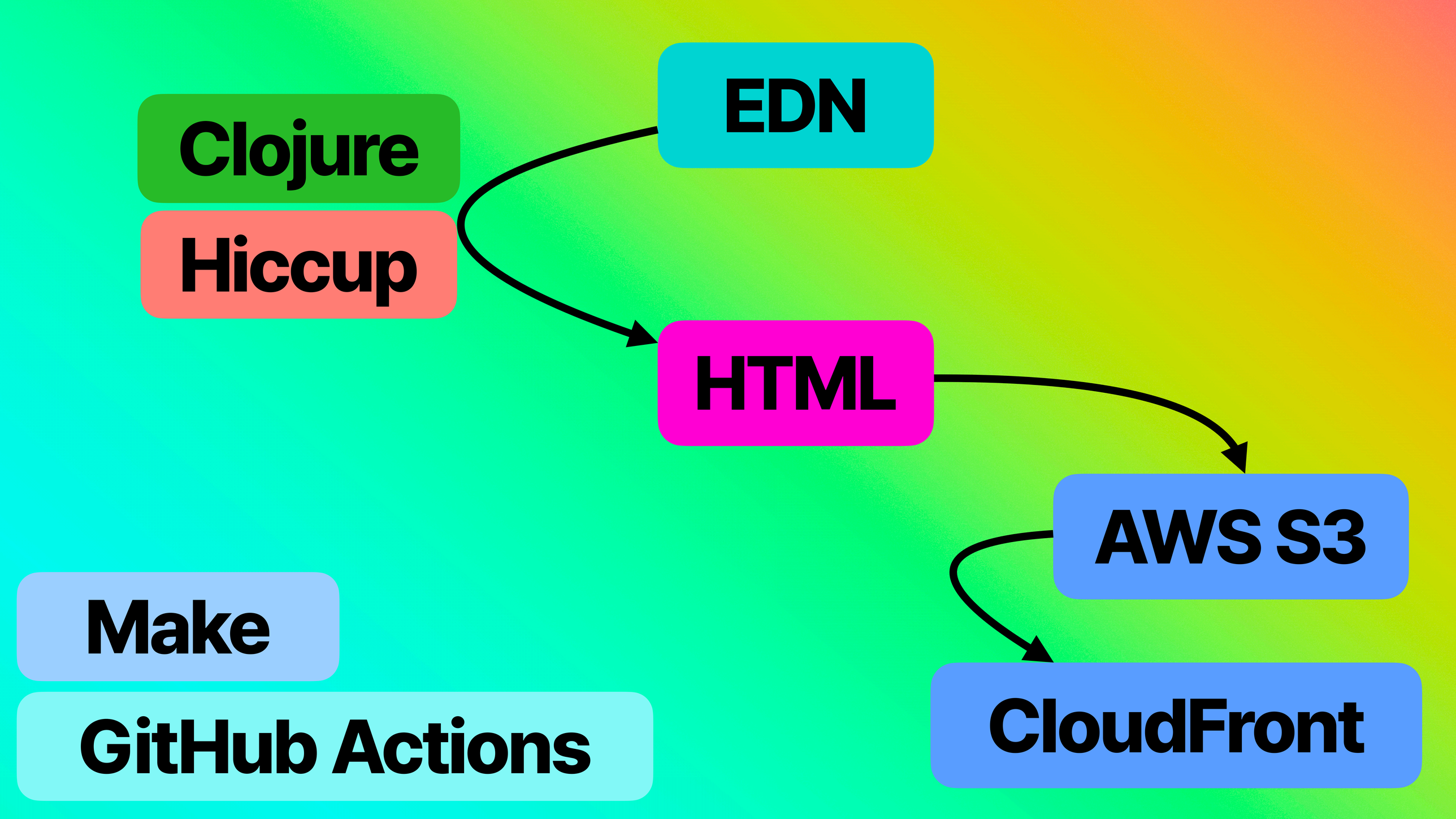
build-index

succeeded 11 hours ago in 3m 32s

Search logs

Set up job	2s
Prepare java	4s
install deps	47s
Setup Clojure	2m 15s
Checkout	1s
Build	**3s**

```
1  ▶ Run clj -M resume.clj > docs/index.html
10 Downloading: org/clojure/clojure/1.11.1/clojure-1.11.1.pom from central
11 Downloading: org/clojure/core.specs.alpha/0.2.62/core.specs.alpha-0.2.62.pom from central
12 Downloading: org/clojure/spec.alpha/0.3.218/spec.alpha-0.3.218.pom from central
13 Downloading: org/clojure/pom.contrib/1.1.0/pom.contrib-1.1.0.pom from central
14 Downloading: hiccup/hiccup/2.0.0-RC3/hiccup-2.0.0-RC3.pom from clojars
```


Clojure
Hiccup
EDN
HTML
AWS S3
CloudFront
Make
GitHub Actions

Q&A

그래서, 이직은 잘 했어요?

Thank you