

# 🔗 La loi d'Atwood rampe vers votre infrastructure

Loi d'Atwood:

Toute application qui peut être écrite en JavaScript sera éventuellement écrite en JavaScript – Jeff Atwood

- L'objectif de cette conférence:
  - Animer, inspirer des idées.
  - Ouvrir une discussions.
  - Partager une histoire autour de la gestion de l'infrastructure.
- JD Caron:
  - Homo Technè

En tant qu'homo technè, nous sommes définis par notre capacité et notre volonté de vivre différemment à travers la fabrication de nouveaux outils. – Reid Hoffman

...je pense qu'une des choses qui nous sépare vraiment des primates c'est que nous sommes des fabricants d'outils... – Steve Jobs
  - Coureur Utilitaire Transpirant l'Espresso 🧤🏠😓☕
  - Bénévole de sentiers

```
; Langages dynamiques
(code-is-data ("data-is-code"))

(loop
  (print
    (eval
      (read))))
```
  - Ce qui me fait du bien:
    - Écrire du code 🧑‍💻, bouger en montagne 🏔️ & manger des plantes 🌱.
  - Projets personnels notables:
    - [Rollance](#): lames pour patins à système de socle universel (Universal Frame System (UFS))
    - [tf2ib](#): système de matchmaking (SourceMod, Python & IRC (communautés [#tf2.pug.na](#) & [#tf2mix](#)))
- L'incident qui a lancé l'écriture de AliaJS (automne 2018):
  - Plateforme (PaaS) qui dégrade en qualité de service; Support technique qui répond que tout va bien 😊

Opinions divergentes sur différents canaux d'information au moment de l'incident: Twitter, [Y Combinator Hacker News](#).
  - Différents outils à l'époque de la panne:
    - Ansible
    - Terraform (OpenTofu?) (500,000 LOC)
    - Chef
    - Pulumi (manquement à ma veille, 400,000 LOC)
    - Kubernetes
    - Jenkins
    - Gitlab CI (~~GitHub Actions~~: 1 an après l'incident)
- Adoption temporaire d'Ansible & migration sur AWS.
- Vision d'écrire un orchestrateur ultra léger (AliaJS):
  - Premier principe, complexité minimale, inspirations: Tiny C Compiler, SQLite
  - Architecture de 1000 lignes, en comparaison aux autres outils qui font plus de 500,000 lignes.
  - Même langage que le projet: Express.js & VueJS.
  - Puissance d'un environnement-langage couplé serré avec le web (JS, HTML, HTTPS, WebSocket, JSON, templating & rendering components).
  - Développement & debug accessible à des développeurs intermédiaires
  - Se comprend en quelques heures
  - Sécurité:

...software—the more you have of it, the more risks you run. – [Daniel Zender](#)
  - Permet des fonctions d'orchestrations complexes comme du déploiement 0 interruptions (downtime) avec support de tests splits, voir code [Lua](#)
    - [AB de Netlify](#) est toujours en beta, depuis [2019](#)
- Avertissement!
  - Pour technologue expérimenté, différence entre AliaJS et les concepts discutés.
- Code:
  - Architecture <https://github.com/jdecaron/aliajs?tab=readme-ov-file#architecture>
  - Migration Redis <https://github.com/jdecaron/aliajs/commit/48cea6df53a15b16bf22c7cb85f97e0460e8e2d3>
  - AB Lua <https://github.com/jdecaron/aliajs/blob/499294de8c2a7fa8575a07d127a1dc1570f9a30c/templates/nginx/locations.ejs#L28-L49>
  - Retry HTTPS DNS lookup <https://github.com/jdecaron/aliajs/blob/499294de8c2a7fa8575a07d127a1dc1570f9a30c/src/deploy.js#L221-L237>
- Demo:
  - AliaJS <https://demo.rotat.io>
    - Code: <https://github.com/jdecaron/aliajs/compare/main...code%40quebec>
  - Render & Upstash <https://aliajs-demo-frontend.onrender.com>
    - Code:
      - Frontend: <https://github.com/jdecaron/aliajs-demo-frontend/blob/main/src/content/docs/index.mdx?plain=1>
      - Backend: <https://github.com/jdecaron/aliajs-demo-backend/blob/main/src/main.ts>
- Shameless plug:
  - Offre de services d'entrepreneur tech 🏠
- Vous êtes bienvenue de me contacter-suivre-collaborer:
  - <https://github.com/jdecaron>
  - [j@aliajs.com](mailto:j@aliajs.com) ou [jdecaron@gmail.com](mailto:jdecaron@gmail.com)
  - <https://www.linkedin.com/in/jdecaron>