



UNE INTRODUCTION AU WEB

SOMMAIRE

Présentation perso

1

Quelques technos
À utiliser...

3

La Scratcheuse

5

C'est quoi être dev
dans une start-up ?

2

Setup un server
en production

4

Questions

6

MAXIME ARBISA

Télécom ParisTech
CTO @LaScratcheuse
Ancien backend
developer @Comet

maxime.arbisa@gmail.com

+33 6 62 37 12 37





1

ÊTRE DEV DANS UNE START-UP



PRODUCT OWNER / MANAGER

- ◇ Ils sont la pierre angulaire d'une start-up
- ◇ Sont au contact des clients et de la boîte pour déterminer les features et les ressources
- ◇ Rassemblent devs, designers, ... pour mener à bien le développement des features



DESIGNER / UX DESIGNER

- ◇ Dessinent les features
- ◇ Mettent en place un 'design system'
- ◇ S'assurent de la cohérence des features dans l'app
- ◇ Pensent un parcours simple et compréhensible aux utilisateurs



TECH JOBS

- ◇ **Backend developers:** s'occupent de la gestion des Bases De Données, de la sécurité, de l'architecture du produit, de sa performance
- ◇ **Frontend developers:** gèrent l'interface, sa fluidité, la compatibilité cross browsers
- ◇ **DevOps:** gèrent l'infrastructure, le déploiement (CI / CD) des apps, la sécurité, downtime à 0%
- ◇ **QA:** qualité du code, mettent en place des tests unitaires, fonctionnels, end to end, ...

DEVELOPPEMENT D'UNE FEATURE

Objectifs du trimestre

Augmenter la rétention chez les freelances

Le produit crée des tickets:

- Permettre un upload des factures
- Permettre de trier les missions en fonction des préférences

Conception

Devs, designers, produits font un brainstorm pour définir comment va être réalisé le ticket:

- Qui fait quoi ?
- En combien de temps ?
- Est-ce bien nécessaire ?

Code

Les devs codent les fonctionnalités et exécutent les tickets.

DEVELOPPEMENT D'UNE FEATURE

Review du ticket

Le produit vérifie que la feature codée répond bien à la problématique de départ.

Allers - retours entre dev et produit

Review du code (QA)

Une fois la feature développée validée, il faut la vérifier:

- Un autre dev review le code
- Le QA s'assure que la nouvelle feature n'a rien cassé

Mise en production

Une fois la feature approuvée, elle peut être mergée sur la prod.

On monitore ensuite la prod pour vérifier que tout s'est bien passé.

CHOISIR SON GESTIONNAIRE DE CODE

GITLAB

- Offre gratuite
- Repo privés illimités
- Utilisateurs illimités
- CI / CD
- Gestionnaire de tickets

BITBUCKET

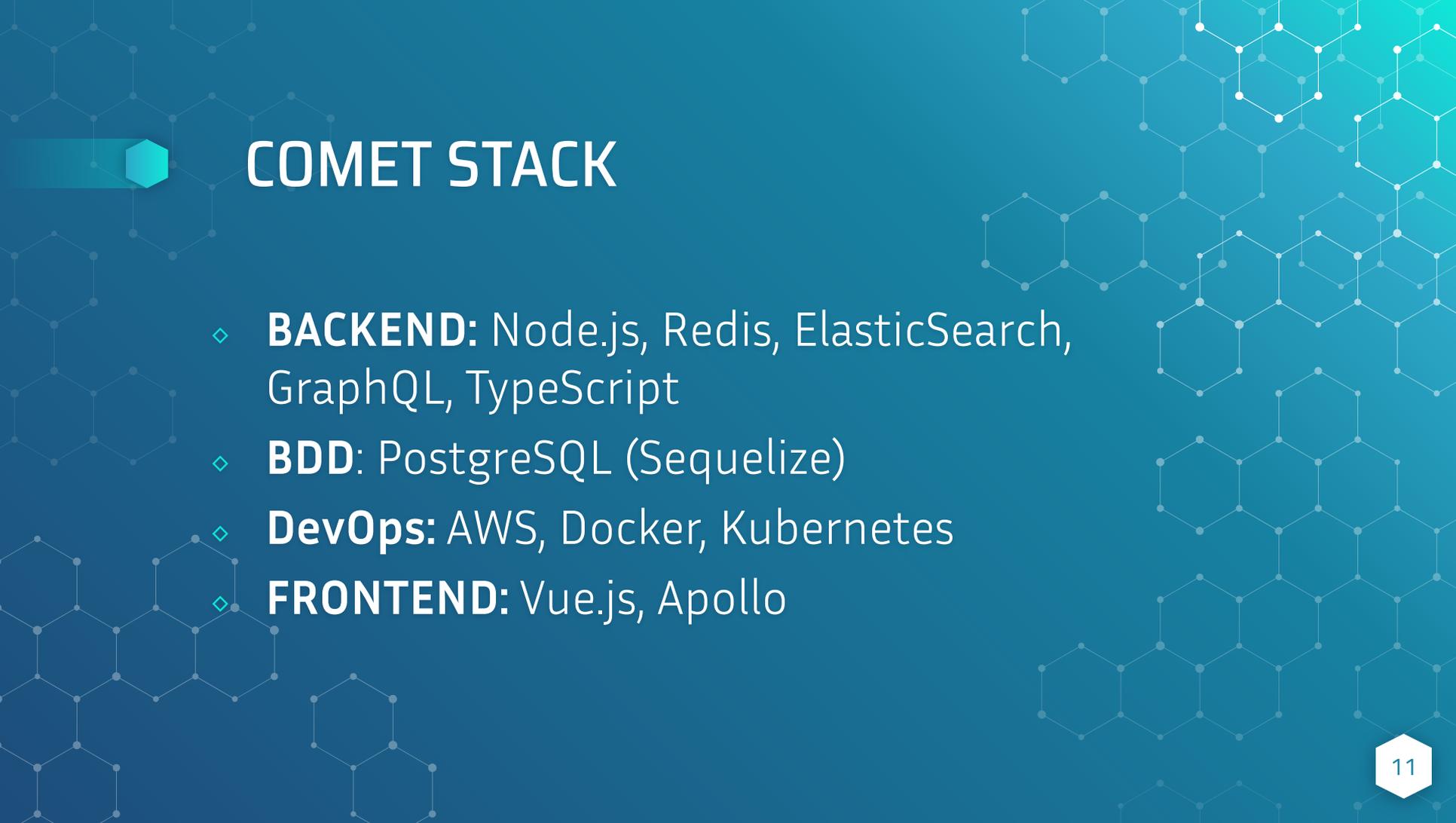
- Offre gratuite
- Repo privés illimités
- 5 users max dans l'offre gratuite
- CI / CD

GITHUB

- Offre gratuite
- Repo privés/publics
- CI / CD
- Gestionnaire de tickets

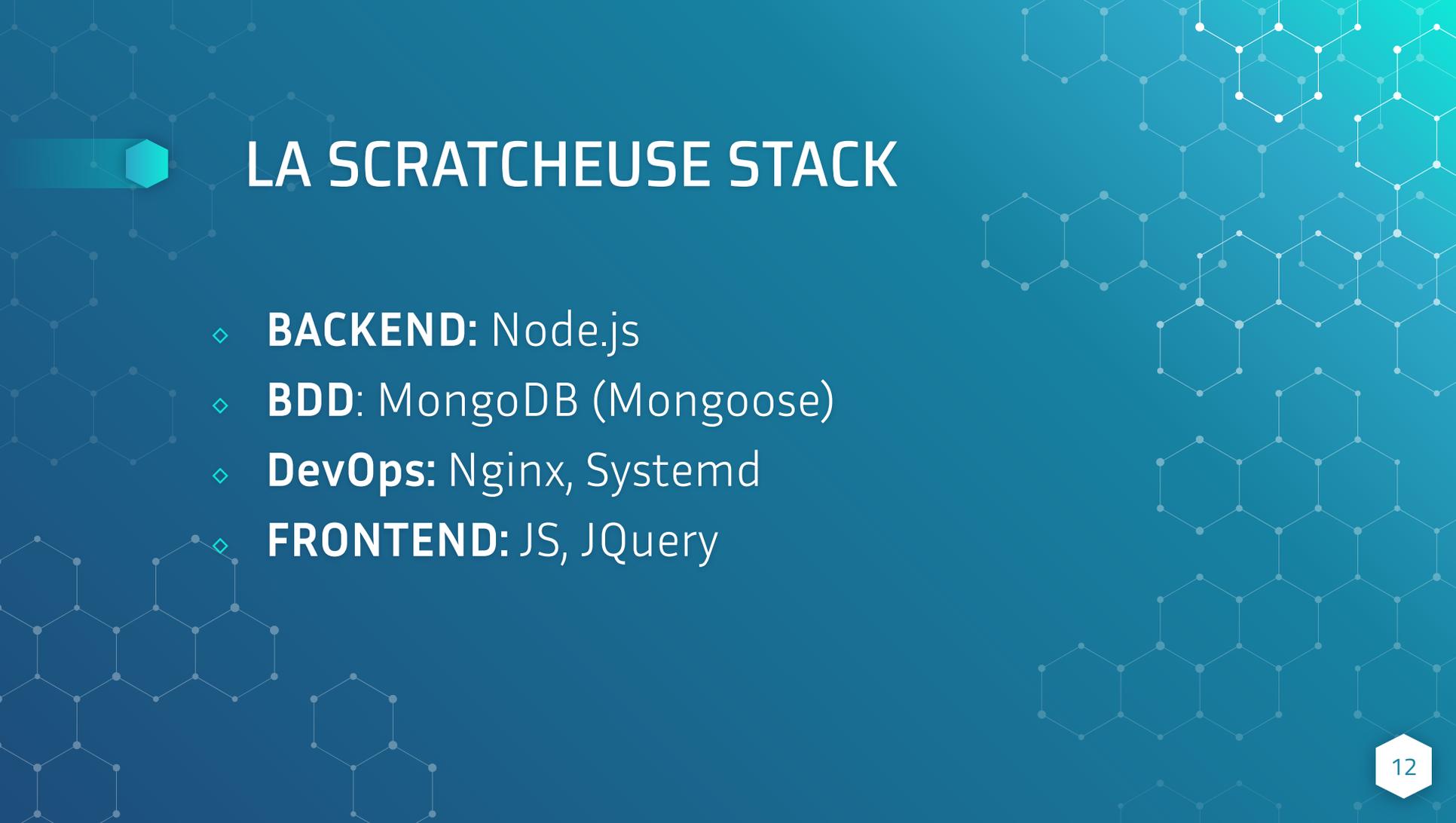
Lequel choisir ?

<https://www.hebergeurcloud.com/github-vs-bitbucket-vs-gitlab-une-bataille-epique-pour-les-pri-de-developpeur/>



COMET STACK

- ◇ **BACKEND:** Node.js, Redis, ElasticSearch, GraphQL, TypeScript
- ◇ **BDD:** PostgreSQL (Sequelize)
- ◇ **DevOps:** AWS, Docker, Kubernetes
- ◇ **FRONTEND:** Vue.js, Apollo



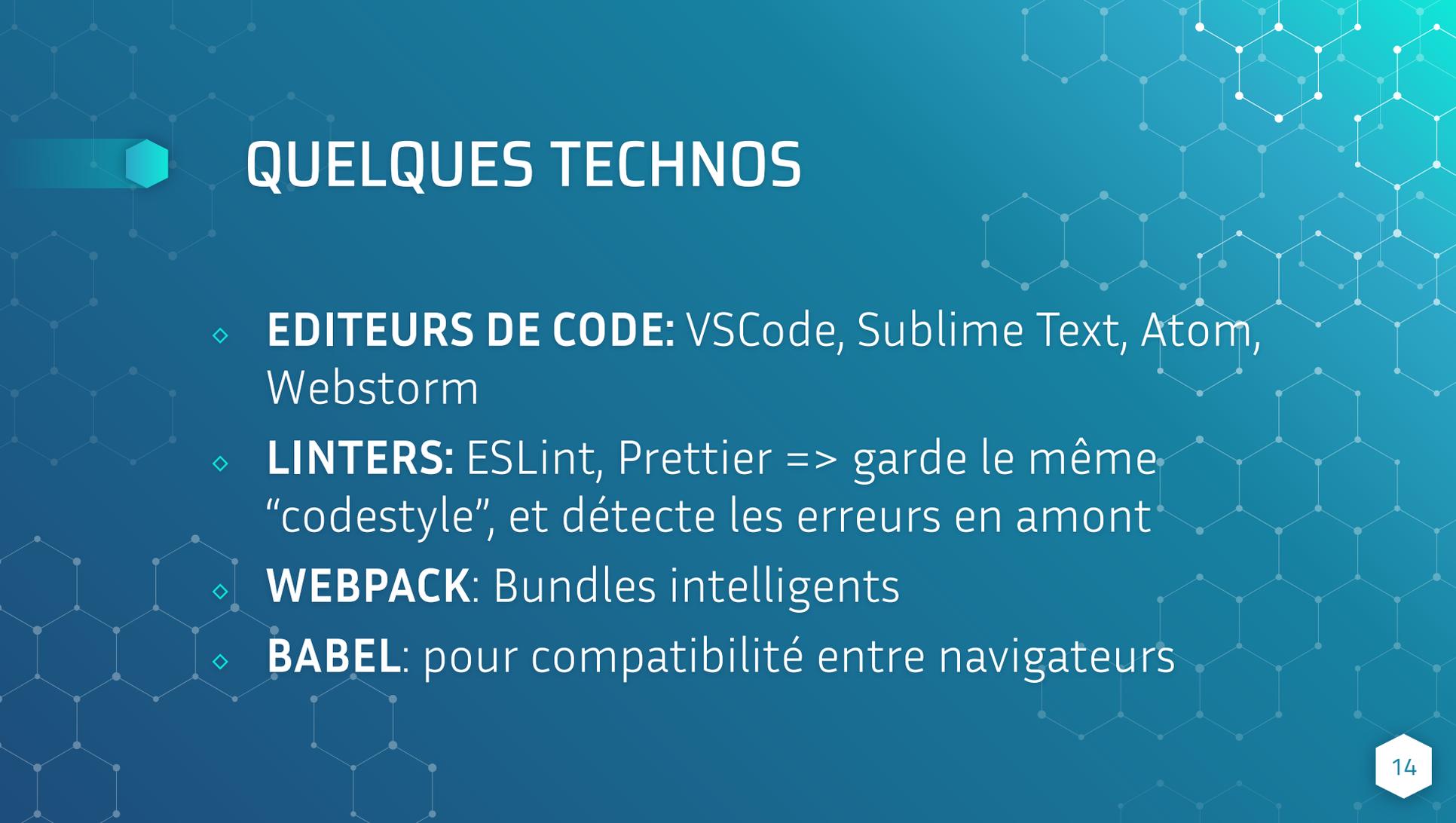
LA SCRATCHEUSE STACK

- ◇ **BACKEND:** Node.js
- ◇ **BDD:** MongoDB (Mongoose)
- ◇ **DevOps:** Nginx, Systemd
- ◇ **FRONTEND:** JS, JQuery



2

QUELQUES TECHNOS



QUELQUES TECHNOS

- ◇ **EDITEURS DE CODE:** VSCode, Sublime Text, Atom, Webstorm
- ◇ **LINTERS:** ESLint, Prettier => garde le même “codestyle”, et détecte les erreurs en amont
- ◇ **WEBPACK:** Bundles intelligents
- ◇ **BABEL:** pour compatibilité entre navigateurs



QUELQUES TECHNOS

- ◇ **MONITORING:** Datadog, Jaeger
- ◇ **TRACKING:** Amplitude, Google Analytics
- ◇ **USER JOURNEY:** Fullstory, Asayer, HotJar



La Scratcheuse

<https://lascratheuse.com>

Créez des instrus facilement



ERROR MANAGEMENT

Bien monitorer les bugs avec
Sentry



POURQUOI SENTRY ?

- ◇ **Gratuit** jusqu'à 5000 erreurs / mois
- ◇ Permet de classifier et d'ordonner les erreurs
- ◇ SDK pour Node.js, Java, React, Vue, Python, ...
- ◇ <https://sentry.io>
- ◇ GitHub example:
<https://github.com/MaximeArbisa/errorManagementWithSentry>



PRODUCTION SETUP

Avec Nginx, Let's encrypt et
Systemd



POURQUOI NGINX ?

- ◇ Serveur web, reverse proxy, load balancer, cache
- ◇ Utilise peu de ressources serveur
- ◇ 2.5 fois plus rapide qu'Apache
- ◇ Utile pour gérer les fichiers statiques devant un process node.js
- ◇ Open source
- ◇ Alternative router: Traefik



LET'S ENCRYPT

- ◇ Fournisseur de certificats TLS gratuit
- ◇ Intégration Nginx rapide et simple
- ◇ Installation:
<https://www.nginx.com/blog/using-free-ssl-tls-certificates-from-lets-encrypt-with-nginx/>



SYSTEMD

- ◇ Gestionnaire de services Linux
- ◇ Gère les problématiques de restart, de dépendances inter services, ...
- ◇ Alternative: PM2



DE L'URL MONSITE.COM AU SERVEUR

- ◇ Faire pointer l'url de votre site vers votre serveur en utilisant **les zones DNS**
- ◇ Record A: nom de domaine -> Adresse IP
- ◇ Record CNAME: nom de domaine -> nom de domaine
- ◇ monsite.com -> A -> 42.82.22.33
- ◇ api.monsite.com -> A -> 42.82.22.33



LIEN GITHUB

- ◇ <https://github.com/MaximeArbisa/setupNginxProduction>

DEPLOYER MON APPLICATION POUR COMMENCER

J'AI UNE API Node.js ?

- Le moins cher est un serveur simple (virtuel ou physique)
- Vous pouvez mettre fichiers statiques, node, BDD sur le serveur
- Suivre le GitHub précédent

Je n'ai qu'une appli frontend ?

- Google Cloud, AWS storage, Vercel, Heroku
- Firebase, Supabase



3

QUESTIONS ?