

Hébergement et déploiement de services

<https://github.com/heig-vd-mvp-course>

[Web](#) • [PDF](#)

L. Delafontaine et V. Guidoux, avec l'aide de GitHub Copilot

Ce travail est sous licence [CC BY-SA 4.0](#).

Plus de détails dans le support de cours

Cette présentation est un résumé du support de cours. Pour plus de détails, consultez le [support de cours](#).

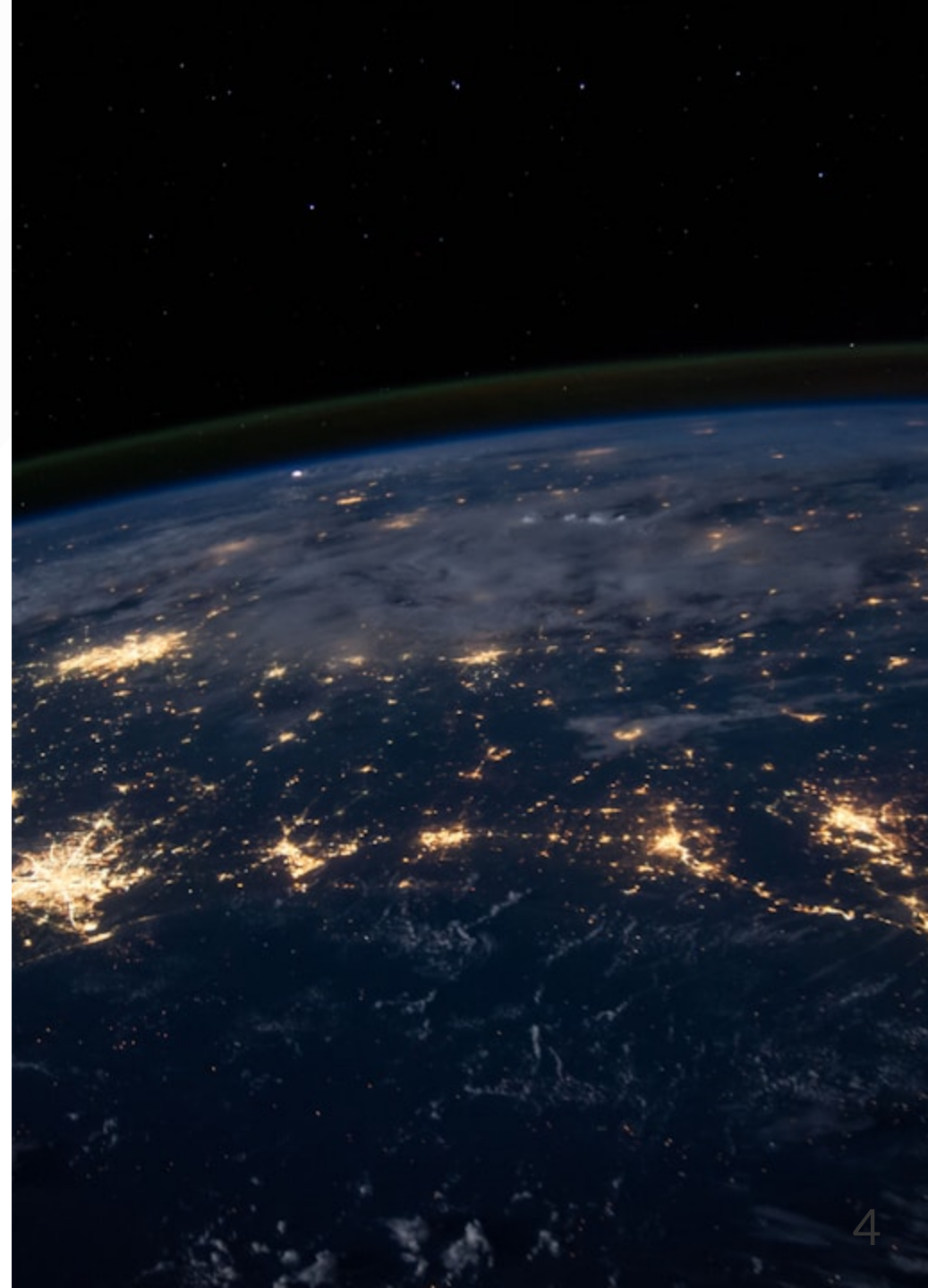
Objectifs

- Décrire comment héberger une application web selon le type de stack technologique
- Décrire où héberger une application web selon le type de stack technologique
- Décrire comment mettre en place un pipeline de CI/CD pour le développement/déploiement continu d'applications



Héberger une application web

- Consiste à rendre une application web accessible sur Internet
- Nécessite un serveur web et un nom de domaine
- Il existe plusieurs types d'hébergement (partagé, VPS, dédié, cloud, PaaS, SaaS, etc.)



Obtenir un hébergement

Il existe plusieurs façons d'obtenir un hébergement :

- Louer un serveur chez un fournisseur d'hébergement
- Utiliser un service d'hébergement cloud
- Utiliser un service d'hébergement PaaS/SaaS
- Votre propre serveur



Obtenir un nom de domaine

- Un nom de domaine est une adresse web (ex. : <https://example.com>)
- Permet d'accéder à votre application web
- Peut être acheté auprès d'un registraire de noms de domaine
- Peut être configuré pour pointer vers votre infrastructure



Déployer une application web

- Consiste à mettre en ligne une application web sur un serveur
- Nécessite de transférer les fichiers de l'application sur le serveur
- Peut être effectué manuellement ou automatiquement



Déployer des sites statiques

- Déjà exploré lors de précédents cours (GitHub/GitLab Pages)
- Hébergement de fichiers statiques (HTML, CSS, JS, images, etc.)
- Extrêmement simple à mettre en place et peu coûteux
- Peu de configuration à faire

Déployer des sites dynamiques

- Contenu généré dynamiquement (ex. : PHP, Node.js, Ruby, etc.)
- Nécessite un serveur d'application pour exécuter le code
- Peut nécessiter une base de données pour stocker les données



Installation et configuration manuelle des outils

- Manière "old-school" de déployer
- Nécessite de configurer manuellement le serveur
- Peut être complexe et long
- Peut nécessiter des compétences techniques avancées



Containerisation des services

- Utilisation de conteneurs pour déployer des applications
- Permet d'isoler les applications et leurs dépendances
- Facilite le déploiement et la gestion des applications
- Peut être effectué à l'aide de Docker ou d'autres outils



Solutions PaaS et SaaS

- Utilisation de services cloud pour déployer des applications
- Permet de se concentrer sur le développement plutôt que sur l'infrastructure
- Très simple à mettre en place mais peut être coûteux
- Une des manières les plus efficaces de déployer des applications

Outils communs pour héberger et déployer une application web

Services d'hébergement

- [Infomaniak](#)
- [Exoscale](#)
- [OVH](#)
- [AWS](#)
- [Google Cloud](#)
- [Microsoft Azure](#)
-



Sites statiques

- [GitHub Pages](#)
- [GitLab Pages](#)
- [Netlify](#)
- [Vercel](#)
- [Cloudflare Pages](#)
- [Firebase Hosting](#)

Sites dynamiques

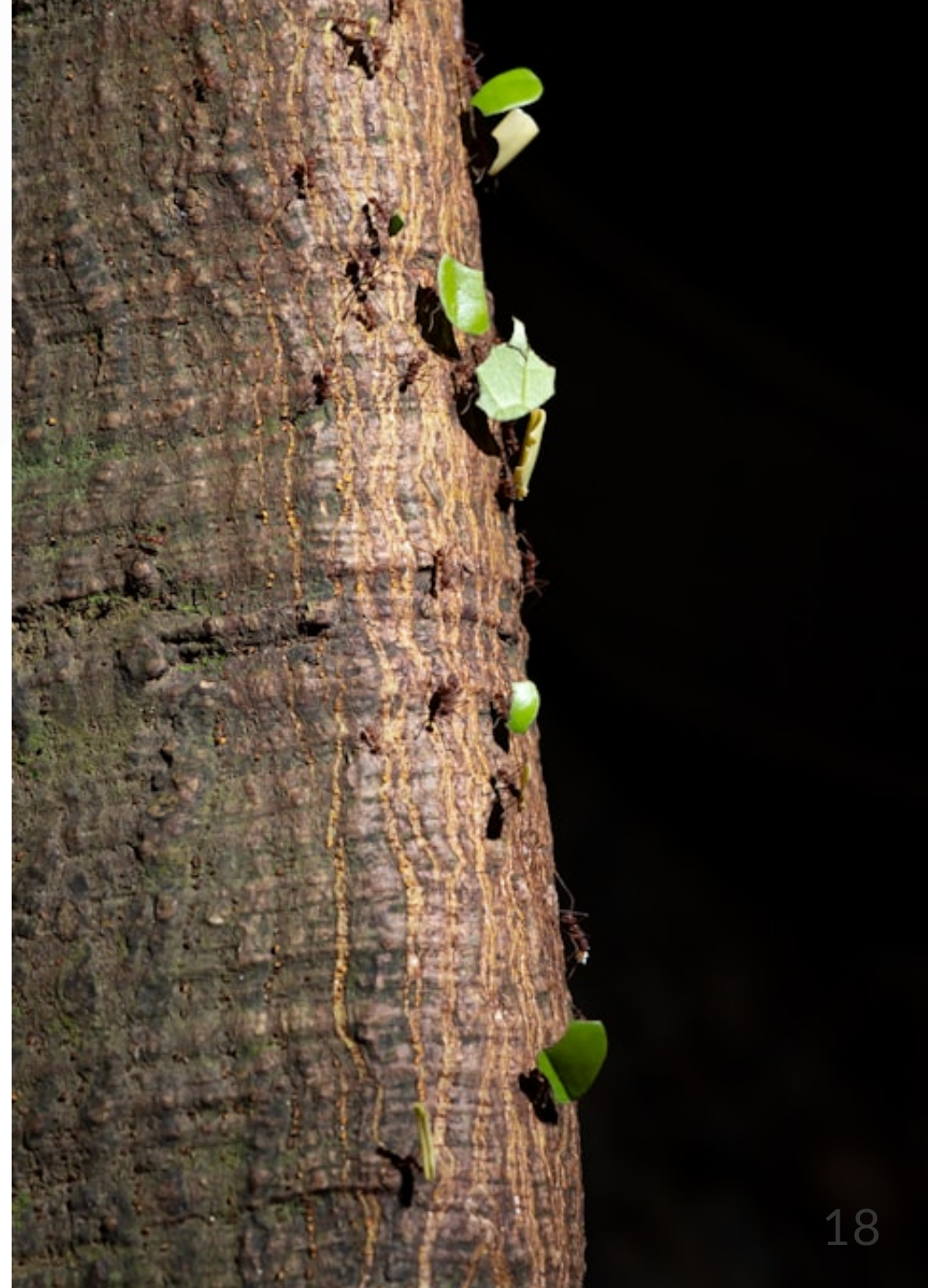
- [Heroku](#)
- [Render](#)
- [Railway](#)
- [Vercel](#)



Mettre en place un pipeline de CI/CD pour la livraison/le déploiement continu d'applications

Que devons-nous faire manuellement (avant même d'utiliser un pipeline) ?

- Review
- Build
- Test
- Containerisation
- Release
- Deploy



Pourquoi automatisons-nous ?

- Minimiser les erreurs humaines
- Gagner du temps
- Standardiser le processus de déploiement
- Documentation du processus de déploiement



Que pouvons-nous automatiser (ou non) ?

TOUT

En fonction du temps et de l'effort que vous êtes prêt à investir.



Formes d'automatisation

Posez-vous la question, quelles ressources avez-vous à disposition ?
Quelle connaissances avez-vous ?

Suivant la réponse, vous allez choisir une forme d'automatisation différente.



Forme *"Push"*

Avantages:

- Simple
- Rapide
- Peu de configuration

Inconvénients:

- Peu flexible
- Pas de contrôle sur le processus de déploiement

Exemples : Heroku, Render, Railway, Vercel

Forme *"Pull"*

Avantages:

- Flexible
- Contrôle total sur le processus de déploiement

Inconvénients:

- Plus complexe
- Plus long à mettre en place

Exemples : GitHub Actions, GitLab CI/CD, Jenkins, Travis CI, Exoscale, AWS, Google Cloud, Azure

Questions

Est-ce que vous avez des questions ?

À vous de jouer !

- Prendre connaissance du [support de cours](#)
- Poser des questions si nécessaire



Sources (1/3)

- [Illustration principale](#) par [Fejuz](#) sur [Unsplash](#)
- [Illustration](#) par [Aline de Nadai](#) sur [Unsplash](#)
- [Illustration](#) par [NASA](#) sur [Unsplash](#)
- [Illustration](#) par [Taylor Vick](#) sur [Unsplash](#)
- [Illustration](#) par [Markus Winkler](#) sur [Unsplash](#)
- [Illustration](#) par [Fabio Sasso](#) sur [Unsplash](#)
- [Illustration](#) par [Michael Dziedzic](#) sur [Unsplash](#)
- [Illustration](#) par [Jan Huber](#) sur [Unsplash](#)

Sources (2/3)

- [Illustration](#) par [Todd Quackenbush](#) sur [Unsplash](#)
- [Illustration](#) par [Marek Piwnicki](#) sur [Unsplash](#)
- [Illustration](#) par [Nikita Kachanovsky](#) sur [Unsplash](#)
- [Illustration](#) par [Stephanie LeBlanc](#) sur [Unsplash](#)
- [Illustration](#) par [Ronnie Schmutz](#) sur [Unsplash](#)
- [Illustration](#) par [Elena Mozhvilo](#) sur [Unsplash](#)
- [Illustration](#) par [Tom Swinnen](#) sur [Unsplash](#)
- [Illustration](#) par [Nik](#) sur [Unsplash](#)

Sources (3/3)

- Illustration par Jumpei Mokudai sur Unsplash