

Introduction au DevOps



Bonjour !

Guillaume Leroy

Intégrateur

pro.guillaume.leroy@gmail.com

[@gleroypro](#)



Credits

Cours réalisé avec Damien Pacaud,
responsable de l'infrastructure chez Teads

1. Historique

Il était une fois dans une galaxie
lointaine, très lointaine...



Avant



Analystes
programmeurs



Administrateurs
système



Analystes programmeurs

- ⬡ Fonctionnalités
- ⬡ Vélocités
- ⬡ Nouveautés technologiques

Administrateurs système

- ⬡ Stabilité (on parle de *uptime*)
- ⬡ Scalabilité
- ⬡ Capacité à réparer rapidement (MTTR)
- ⬡ Maîtriser le changement
- ⬡ Mesurer les impacts

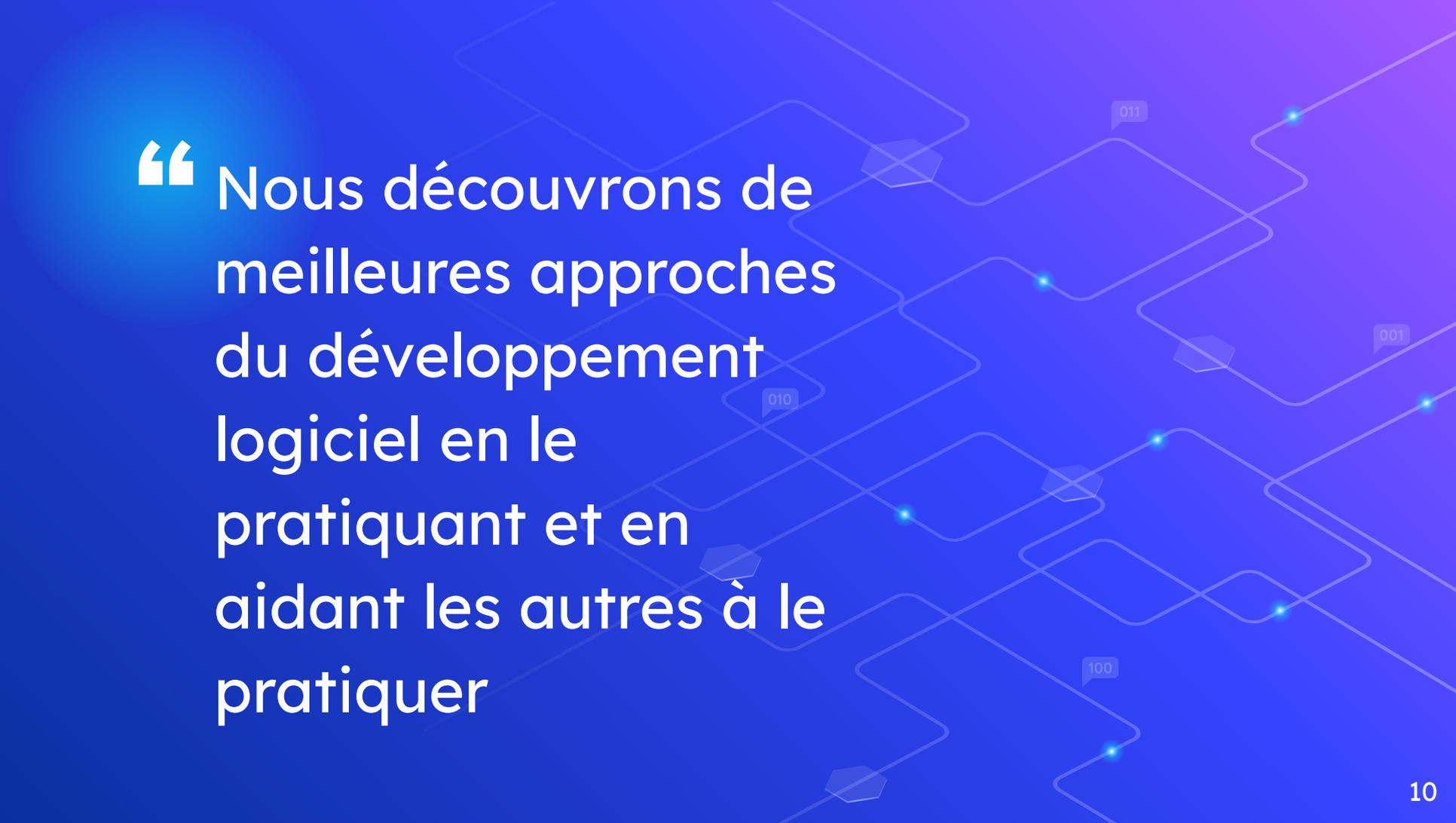
Problèmes

- ⬡ Mauvaise communication
- ⬡ *Chez moi ça marche!*
- ⬡ Moins de responsabilité chez un programmeur

2001

Manifeste Agile





“ Nous découvrons de meilleures approches du développement logiciel en le pratiquant et en aidant les autres à le pratiquer

Cycle de vie d'un projet



Et techniquement parlant ?

- ⬡ Plus de tests automatisés !
- ⬡ Démocratisation de l'intégration continue
- ⬡ Faciliter le déploiement des applications

Et humainement parlant ?



Administrateurs
système



2009

Velocity Conf 09 (1)



2. DevOps mindset

Culture du DevOps





“ Culture
Automation
Measurement
Sharing

Culture

- Casser les murs entre les équipes
- Accepter le changement
- Transparence

Automation

- ⬡ *Automate all the things!*
- ⬡ *One-click build*
- ⬡ *One-click deploy*

Measurement

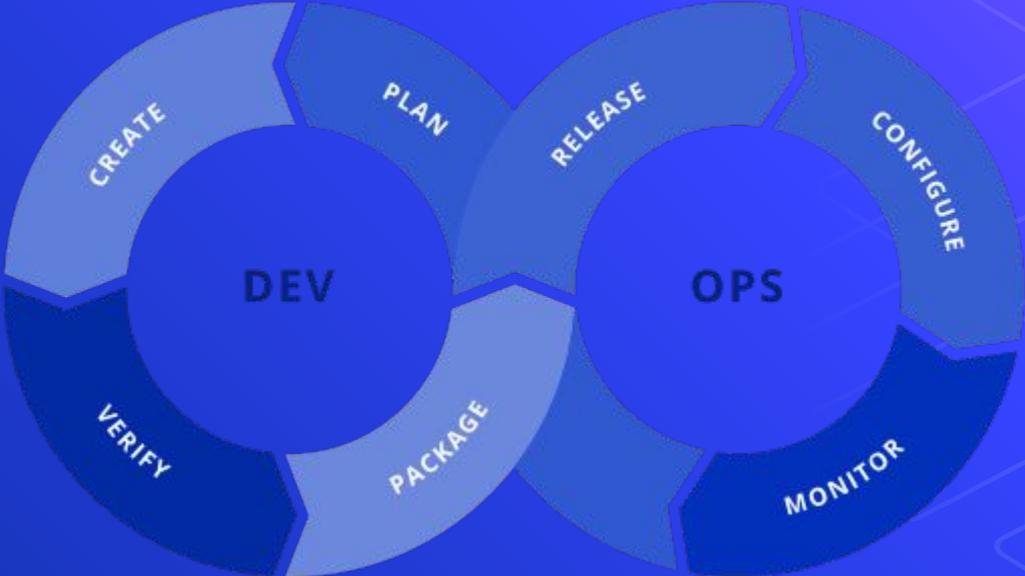
- *Monitor all the things!*
- Ce qui ne peut être mesuré n'existe pas
- Distribution

Sharing

- ⬡ Le code
- ⬡ Les outils
- ⬡ Les connaissances



DevOps Loop



3. Les outils

Cloud, conteneurisation, IaC...



Docker

Conteneurisation



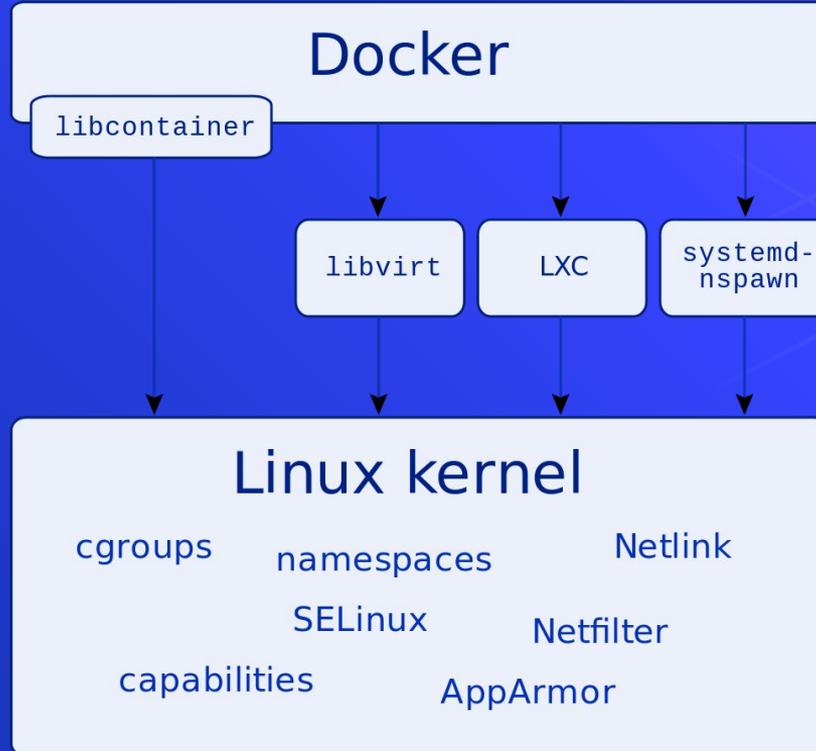
Conteneur

- Virtualization au niveau de l'OS
- Isolation des accès et des ressources d'**un** processus
- Chroot / CGroup / Namespace

Docker

- ⬡ A facilité la création de conteneur
- ⬡ Facilite la création d'environnement

Architecture



Concept

- Image
- Conteneur
- Réseau
- Volume



Orchestration

- ⬡ Management du cycle de vie des conteneurs
- ⬡ Partage des ressources d'un cluster

Cloud



Solutions

IaaS	PaaS	SaaS
Infrastructure as a Service	Platform as a Service	Software as a Service
Fournit des machines virtuelles	Orchestre des conteneurs	Boîte noire
AWS, GCP	Heroku, OpenShift	-

Régions



VPC

- Réseau privé virtuel
- Un VPC par région
- Un sous réseau par zone



Instance

- ⬡ Configuration de l'OS
- ⬡ Configuration des ressources



Load Balancer

- Composant managé
- Équilibre le trafic entre plusieurs instances
- Ajoute / retire des instances de manière autonome en fonction du trafic

Élasticité

- ⬡ Groupe d'instances
- ⬡ Ajuste le nombre d'instance
- ⬡ Condition de mise à l'échelle configurable



Object storage

- ⬡ Espace de stockage infini
- ⬡ Temps d'accès très lents



Base de données

- ⬡ Configuration du SGBD
- ⬡ Haute disponibilité

Coût

- ⬡ On paye ce que l'on consomme
- ⬡ Assez complexe
- ⬡ Vérifier régulièrement les factures

IaC

Infrastructure as Code



IaC

- Décrire une infrastructure dans du code
- Automatisation
- Reproductibilité
- Versionable
- Relecture
- Documentation

Choix

Spécifique au vendeur	Généraliste	Personnalisé
Cloud formation, Deployment manager	Terraform	Bash, Go, Perl...

4. L'automatisation

CI/CD



CI

Continuous Integration



Intégration...

- Mettre en commun les différents travaux des développeurs
- S'assurer de la bonne exécution du build et des tests

... continue

- ⬡ Automatiser l'exécution du build et des tests
- ⬡ Réaliser cette opération à chaque changement apporté au code

Objectifs

- Réduire les impacts des bugs
- Améliorer la qualité et la quantité des tests
- Augmenter la fréquence des releases
- Réduire les risques

CD : Continuous Deployment

- Déploiement automatisé
- Éviter les arrêts de service
- Automatisation des rollbacks si une erreur survient

Stratégies de déploiement

Rolling update	Blue/Green	Canary
Incrémentale, serveur par serveur	Nouvelle version dans un groupe de serveur, redirection du trafic	Nouvelle version dans un groupe de serveur, redirection du trafic progressive

Objectifs

- Limiter les impacts
- Rendre les mises en production anecdotique
- Réduire le *time to market* d'une fonctionnalité

Exemple de pipeline

- Build
- Exécution des tests unitaires
- Conteneurisation
- Lancement d'un conteneur dans un environnement
- Exécution des tests d'intégration

Exemple de pipeline

- ⬡ Exécution des tests de non régression
- ⬡ Release
- ⬡ Déploiement sur le serveur de recette
- ⬡ Déploiement sur le serveur de préproduction
- ⬡ Déploiement en production

Liste non exhaustive d'outils

- Jenkins
- GitLab CI
- Travis
- GitHub Actions
- ...

5. Provisionnement

Un peu d'admin sys'



Définition

- Installation des logiciels
- Configuration
- Mise à disposition

Manuel

- ⬡ Risque d'erreurs
- ⬡ Non uniforme sur toutes les machines
- ⬡ Pas de documentation
- ⬡ Ne passe pas la mise à l'échelle

Outils de provisionnement

- Deux types : avec et sans agent
- Deux manières : *pull* et *push*
- Deux philosophies : descriptif et déclaratif

Quelques outils...

- ⬡ Salt
- ⬡ Puppet
- ⬡ Ansible



5. Monitoring

S'assurer qu'un service en production
reste en production



Monitoring

- Métriques
- Logs
- Dataviz

Monitoring

- Systeme & reseau
- Logs applicatif
- Dataviz

Alerting

- ⬡ Passage d'un seuil
- ⬡ Différents niveaux d'alertes
- ⬡ Email, Slack...



Base de données temporelle

- ⬡ Modèle de données basé sur le temps
- ⬡ Temps-valide
- ⬡ Temps-transaction
- ⬡ Donnée bi-temporelle

Quelques outils

- ⬡ Elasticsearch Logstash Kibana (ELK)
- ⬡ Prometheus
- ⬡ Nagios
- ⬡ Grafana
- ⬡ ...

6. Futur

Nom de Zeus...



Serverless

- ⬡ Pas de déploiement
- ⬡ Juste du code
- ⬡ Contraintes sur les ressources fortes

Ops

- ⬡ Développement d'outils
- ⬡ Partage des responsabilités



Merci à tous !

Des questions ?

Par mail :

pro.guillaume.leroy@gmail.com

