



VI. PLATFORM AS A SERVICE

Stéphane WOUTERS

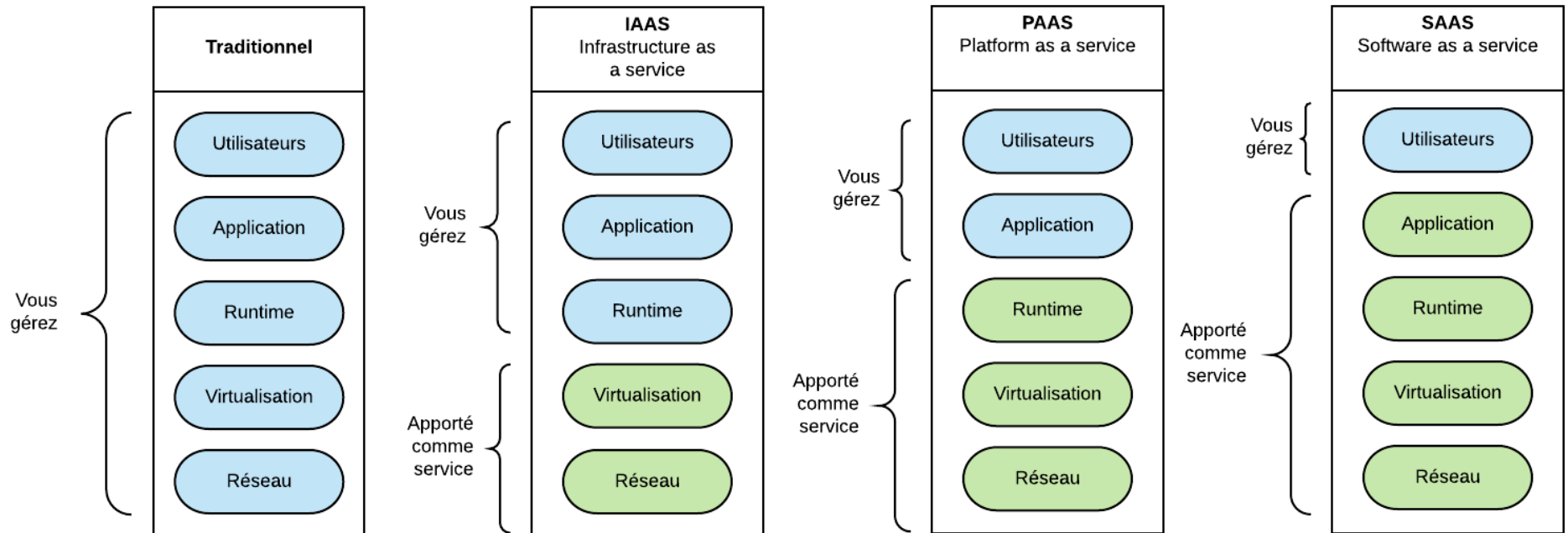
École WIS 2019/2020



PAAS

PLATFORM AS A SERVICE

PAAS EST UN NIVEAU DE CLOUD



QU'EST CE QUE LE RUNTIME ?

Environnement d'exécution en français

**Ensemble des programmes permettant
d'exécuter une application**

Exemple : WAMP sur Windows

DIFFÉRENCE AVEC IAAS

En IAAS, on obtient une VM "vierge"
On installe soi même le runtime (PHP, node.js,
serveur ftp...)

AVANTAGES DU RUNTIME AUTO-GÉRÉ

- Les versions maintenues par l'opérateur
 - Les mises à jour de sécurité sont réalisées par l'opérateur
- L'opérateur a l'expertise pour configurer et optimiser le runtime correctement

FACTURATION EN PAAS



La facturation est réalisée de la même manière qu'en IAAS :
L'utilisateur choisit les machines dont il a besoin. **Il est facturé à l'utilisation de ressources**

Pico	Nano	XS	S	M	L	XL	2XL	3XL
1 CPU	1 CPU	1 CPU	2 CPUs	4 CPUs	6 CPUs	8 CPUs	12 CPUs	16 CPUs
256 MB	512 MB	1 GB	2 GB	4 GB	8 GB	16 GB	24 GB	32 GB
4.50€ / Month	6.00€ / Month	14.40€ / Month	28.80€ / Month	72.00€ / Month	144.00€ / Month	288.00€ / Month	576.00€ / Month	1152.00€ / Month
ADD	ADD	ADD	ADD	ADD	ADD	ADD	ADD	ADD

Tarifs de CleverCloud

Le coût est souvent plus élevé qu'en IAAS pour les mêmes ressources. C'est expliqué par le service supplémentaire apporté.



AUTO SCALLING

Les PAAS sont compatibles avec l'auto scaling.
Mais il doit être configuré par l'utilisateur.

COMPATIBILITÉ



L'application hébergée en PAAS doit être **compatible Cloud** et suivre les normes d'hébergement modernes. Souvent l'application doit être légèrement modifiée pour être compatible avec le PAAS (création d'un script d'initialisation, ajout de dépendances...)



Pour profiter de l'auto scaling, l'**application hébergée doit être répliquable** (utiliser du stockage cloud par exemple, voir cours sur S3)

DÉPLOIEMENT

En PAAS on essaie de faciliter au maximum le déploiement.

L'idée est de permettre aux développeurs de déployer à tout moment



Plusieurs méthodes :

- **Connecté au GIT**
- Par dépôt **FTP**

platform.sh 

N'ayez plus peur de déployer le vendredi

L'automatisation de votre workflow vous assure la totale fiabilité de vos nouveaux développements. Ainsi, aucun risque d'erreur : vous permettez à vos équipes de déployer chaque nouvelle fonctionnalité plus rapidement et facilement...même le VENDREDI !

Avantages	Inconvénients
"You Write Code. We Run It." L'utilisateur dépose son code sur le PAAS, il n'aura jamais besoin de se connecter à la machine	Choix du runtime limité. En cas de besoin très spécifique (extension PHP particulière, langage très peu connu), le PAAS ne pourra pas être utilisé
Orienté à usage des développeurs. Ne demande aucune connaissance en administration serveur	Plus coûteux qu'en IAAS
Déploiement simplifié et automatisé	On fait confiance au prestataire pour la gestion serveur. Aucun recours possible en cas de panne.
	L'application doit être compatible "Cloud"

ET LES HÉBERGEMENT WEB MUTUALISÉS ?

- Les serveurs sont auto gérés
- & On dépose le code par FTP



C'est du PAAS ?

Oui et non :

- Auto scaling non disponible
- Souvent compatible uniquement PHP
- Souvent trop limité en ressources pour les applications plus lourdes (symfony...)

MARCHÉ



clever cloud



platform.sh



heroku



PAAS

MISE EN PRATIQUE

AVEC CLEVER CLOUD

<http://bit.ly/2GOqnER>



LES CONTAINEURS

Une révolution dans la répartition des
responsabilités développeurs / opérateurs

CLASSIQUEMENT

RESPONSABILITÉ DES PARTIES



Développeurs

Écrivent le code source
de l'application



Opérateurs

Récupèrent le code source,
configurent les serveurs (VM,
runtime...) et assurent la
maintenance (performances,
scalabilité...)

Que ce soit en IAAS, PAAS, SAAS... Les
responsabilités restent les mêmes

PROBLÈME...

Les développeurs ne gèrent pas le runtime mais
ont parfois des besoins spécifiques



Développeur :

*“ Mon application nécessite PHP 5.4
uniquement et ImageMagick-6.9.10-8*

COMMENT DÉCRIRE LE RUNTIME ?

Écrire les spécifications dans un fichier texte
quelque part sur le projet...



```
1 - Apache 2.4
2 - PHP 5.3
3 - ImageMagick-6.9.10-8 (attention compiler uniquement avec RSVG)
4 - Memcached
5 - etc...
```

Tout le monde (développeurs, opérateurs) doit
respecter ces versions. Aucune garantie n'est
possible



UNE SOLUTION À CE PROBLÈME

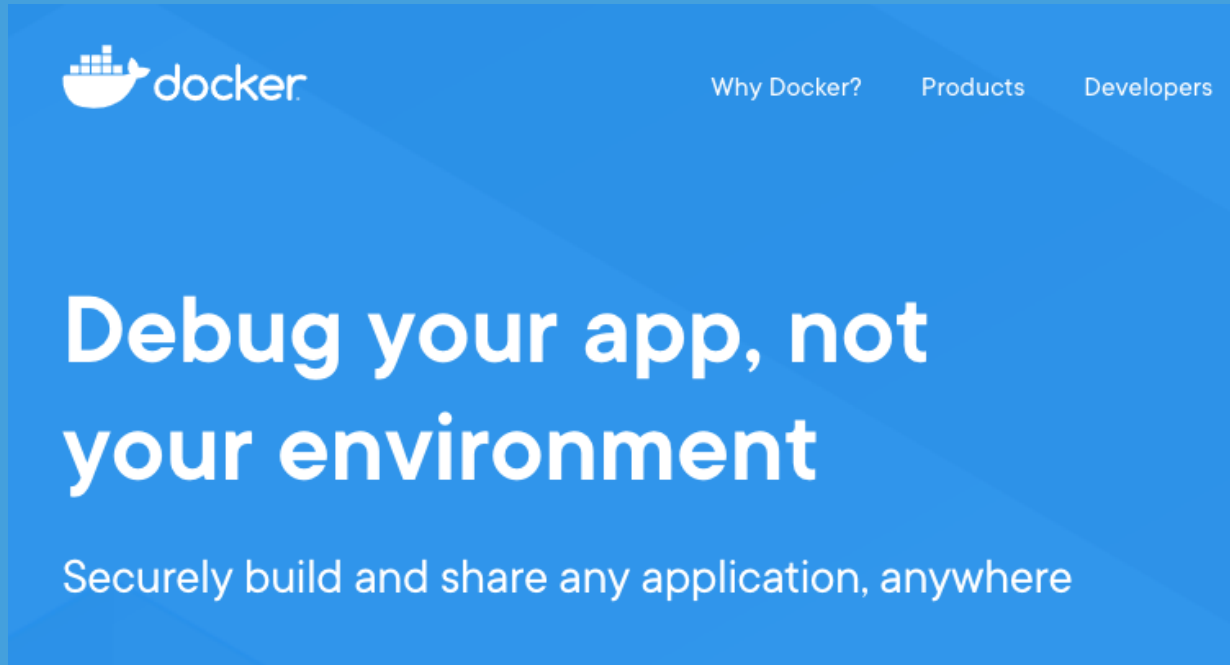
LES CONTAINEURS

Le développeur crée une "image" de son application qui embarque le runtime ET le code source

Son application devient une boîte noire hébergeable sur n'importe quel serveur sans connaître son contenu (langages, code source...)

C'est un changement de responsabilité entre les développeurs et les opérateurs

La technologie la plus populaire pour créer des conteneurs :



DOCKER



DÉMO DOCKER

LES AVANTAGES SONT NOMBREUX

- Environnements (local/dev/recette/prod) ISO
 - Si ça fonctionne en local => ça fonctionne en production
- Facilité des déploiements
- Gain de temps sur les échanges entre développeurs/opérateurs sur le runtime

MAIS CRÉE DE NOMBREUX PROBLÈMES PHILOSOPHIQUES



Les opérateurs perdent en visibilité sur les applications qu'ils hébergent

Ils ne peuvent plus configurer le "runtime" et affiner la configuration sur mesure

Ils ne peuvent plus assurer la qualité de services qu'ils souhaiteraient

CONTAINEURS ET PAAS

En résumé...

L'usage de container se rapproche du principe du PAAS : On pose une application sur le service du prestataire, il s'occupe de l'exécuter

Mais en général les PAAS déconseillent l'usage des conteneurs car ils perdent en contrôle sur les applications