PILOT SYSTEMS

Développer dans le Cloud

Gaël LE MIGNOT
Pilot Systems

27 juin 2020

Déroulé

Introduction

Présentation

L'intervenant : Gaël Le Mignot

- Diplômé de l'Epita (spécialité IA) en 2002
- Contributeur au projet GNU/Hurd
- Développeur Python, C, C++
- Administrateur système Debian GNU/Linux

Pilot Systems

- Société de service
- Développement d'applications Web
- Hébergement / infogérance
- Conseil, formation, ...

Quelques clarifications

Qu'est-ce que le cloud ?

- Vaste sujet : cloud public, privé, ...
- Des machines virtuelles en accès distant
- Idéalement redimensionnables

Précisions sur le périmètre

- Développement... de projets Web
- Taille des équipes : petites à moyennes

Pourquoi développer dans le cloud ?

Pourquoi développer dans le cloud ?

Plus proche de la prod

Au niveau matériel

- Même type de matériel
- Même connectivité réseau

Au niveau logiciel

- Mêmes versions d'OS et de logiciels
- Plus facile de mettre du SSL (Let's Encrypt, wildcard)

Accessible de l'extérieur

Meilleur collaboration

- Intervention d'un collègue sur son instance
- URLs visibles par tout le monde

Mais aussi en solo...

- Consultation mobile
- Retour de service API (ex : paiement en ligne)
- Service d'analyse (ex : validateur HTML/CSS)

Plus de liberté

Compatible avec le nomadisme

- Ordinateur de bureau + laptop
- Télétravail, parfois soudain
- Interventions ponctuelles depuis des congés

Intégration de nouvelles personnes

- Plus simple de créer de nouvelles instances
- Possibilité de prendre de l'avance

Ajustement de ressources

- Stockage : grosses migrations
- RAM / CPU : gros traitements

Plus de sécurité

Fiabilité accrue du Cloud

- Matériel en panne ? Redondance.
- Effacement accidentel ? Sauvegardes.
- Crash de disque dur ? RAID.

Moins de fuite de données

- Matériel volé
- Intervenant sur le départ

Inconvénients

Connexion Internet

- Nécessite une bonne connexion Internet
- Impossible de travailler hors ligne
- Aspects firewall/VPN à gérer

Compétences requises

- Compétences en admin/scripting ou Docker
- Pour la mise en place des instances, pas pour l'utilisation

Compatibilité IDE

- Peut poser problème dans certains cas
- Cette partie sera développée plus tard

Risques

Disponible à l'extérieur

- Indexation sur les moteurs de recherche
- Exploitation de failles de sécurité
- Solutions : VPN, `robots.txt`, `htaccess`.

Envoi de mails

- En général dans le Cloud on peut envoyer des mails
- Risque d'envoyer des mails à des vrais utilisateurs
- Solution : redirection, applicative ou système

Aspects financiers

Coûts supplémentaires

- Chez Pilot Systems, coût indirect
- Souvent, coût direct, par instance

Mais économies..

- En temps, souvent bien plus précieux
- Sur le matériel nécessaire pour les devs

Ce qui ne change pas...

Git

- Il faut toujours un git
- On `commit` / `push` / `pull` sur le serveur
- Rien ne change pour les branches

Autre

- Peer review
- Système de tickets
- Documentation
- CI / CT

Comment développer efficacement dans le Cloud

Gestion des bases de données et des instances

Instances individuelles ou partagées

- Une instance par développeur : pour les gros projets
- Une seule instance : pour les petits projets
- Une instance par branche
- Solutions hybrides

Gestion des bases de données

- BDD partagée : plus simple, moins coûteux
- Migrations de schéma de BDD
- Solution hybride
- Synchro prod => dev (semi)-automatique

Comment éditer à distance

Édition dans le terminal

- Emacs / vi / nano
- Pas forcément installé partout
- Pas confortable, copier-coller compliqués, ...

Solutions intégrées à l'éditeur

- Tramp dans Emacs
- `scp://user@host:/path/to/file`
- Emacs donc historique persistent, ...
- D'autres éditeurs / IDE offrent ça

Comment éditer à distance : sshfs

Exemple

```
~ $ mkdir tb
~ $ sshfs testbuster.pilotsystems.net:/ tb
~ $ cat tb/etc/hostname
testbuster
~ $ grep address tb/etc/network/interfaces
address 91.220.85.118
```

Précisions

- Donne un vrai FS, utilisable par tout logiciel
- Linux seulement : s'appuie sur fuse
- Disponible en UI (ex : Nautilus)

Comment éditer à distance : autres options

Autres OS

- MacOS X : cyberduck
- Windows : port de `sshfs` existe

Autres options

- Unix : montage NFS
- Windows : SMB / samba
- Tous les OS : Webdav

SSH efficace : agent

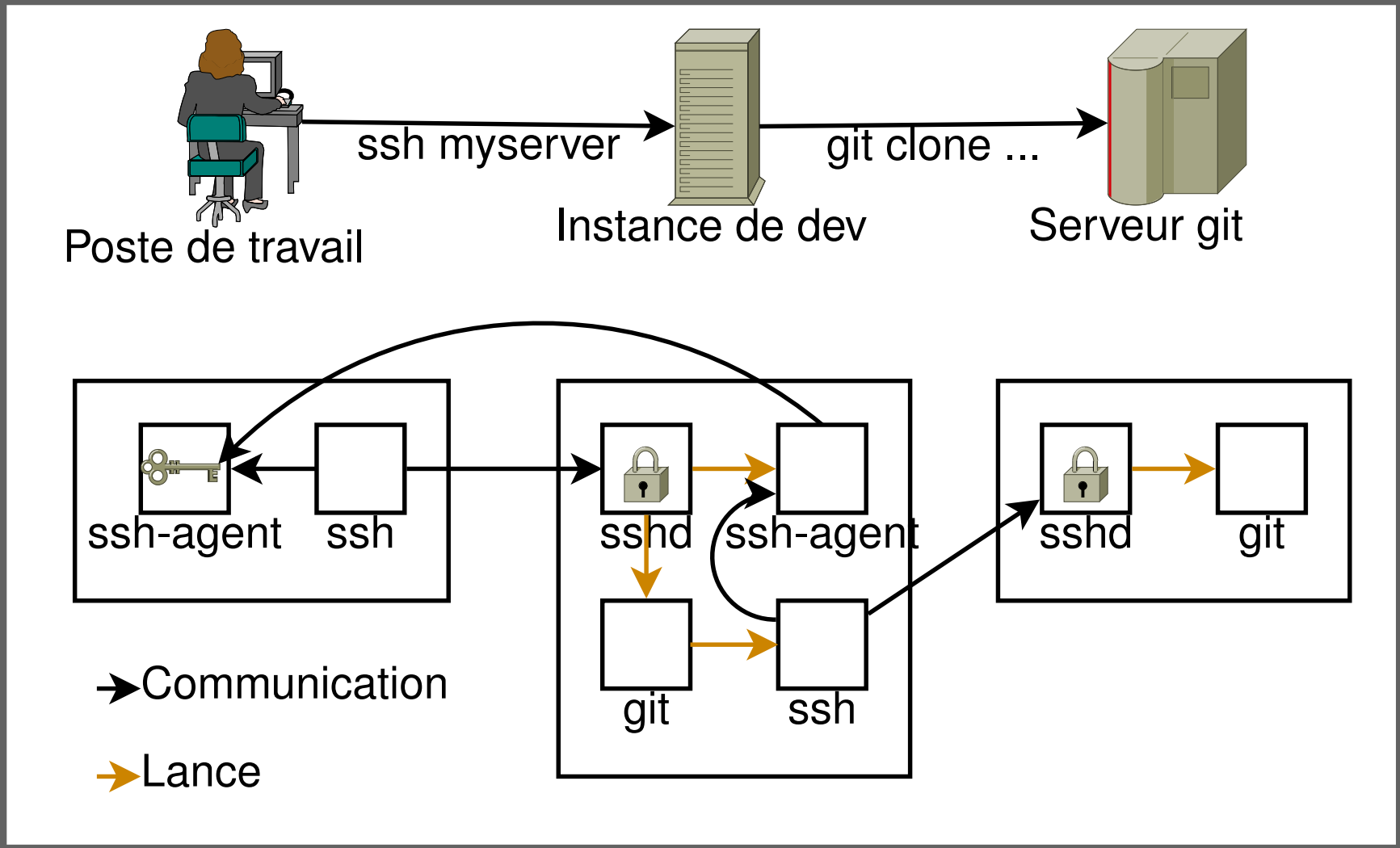
Clefs SSH

- Clef publique : diffusée sur les serveurs
- Clef privée : protégée par mdp, ne pas diffuser
- Génération : `ssh-keygen`

Agent SSH

- Lancé par la session (ex : Gnome)
- Ou à la main : `eval $(ssh-agent)`
- Garde les clefs déverrouillées pour la session
- Capable de retransmettre les demandes (ex : git)

SSH efficace : agent



SSH efficace : configuration

Utilité

- Spécifier l'IP, le nom d'utilisateur
- Rebondir

.ssh/config

```
Host bastion
User admin
Hostname 42.42.42.42

Host varnish
User admin
ProxyCommand ssh bastion nc -q0 10.42.42.42 %p
```

Screen

Screen c'est quoi ?

- Multiplexeur de terminal
- Notion d'attachement et déttachement
- Plusieurs terminaux en un

Screen 101

- Lancement : `screen`, `screen -Rd`
- Raccourcis : `C-a d`, `C-a c`, `C-a C-a`, `C-a <0-9>`

Alias pratique

```
sss() {
  ssh $@ -t screen -Rd
}
```

Les containers

Les containers

Création d'instances

Création manuelle

- Ok si très peu de développeurs
- Vite pénible et consommateur de temps

Via script

- Script de déploiement ou de copie
- Plus ou moins compliqué suivant le projet
- Configuration DNS, frontal Web, SSL, ...

Les containers

Container 101

- Mini-VM, sans noyau
- En réalité namespace
- Image (ex : Debian 10) et container (instance)
- Layers (volatile) et volumes (persistants)

Comment faire ?

- `Dockerfile`
- Images déjà existantes
- `docker-compose` pour des projets plus complexes

Code as data ?

Le problème

- En dev, le code change beaucoup
- Recompiler l'image à chaque modif : trop lourd

Solution

- Utiliser un volume pour le code
- Pour le container, le code sera donc des données
- En dev uniquement, bien sûr

Limitations

- Introduit des différences entre dev et prod
- Mais différences inévitables (bdd, ...)
- Images modulaires

Utilisation en local et dans le cloud

En local...

- Plus simple via les containers
- Mais il reste la problématique des données
- Et tous les autres aspects évoqués

Hybride

- Avoir des instances en local et dans le cloud
- Par exemple : développeur à faible connexion, ou travaillant dans le train

Conclusion

Conclusion

Contacts

Pilot Systems

- Site Web : `https://www.pilotsystems.net/`
- Téléphone : +33 1 44 53 05 55

Gaël Le Mignot

- Contact technique
- Mail : `gael@pilotsystems.net`

David Sapiro

- Contact commercial
- Mail : `david@pilotsystems.net`
- Whatsapp : +212 6 61 801 861

Conclusion

Merci de votre attention.
Des questions ?