Breizh C@mp
Mix de technologies

Merci Docker, tu m'as aidé à respecter les SLA de mon service Support Client.

#partage #docker #supportclient

Hung BUI - @bqh35

https://about.me/bqh

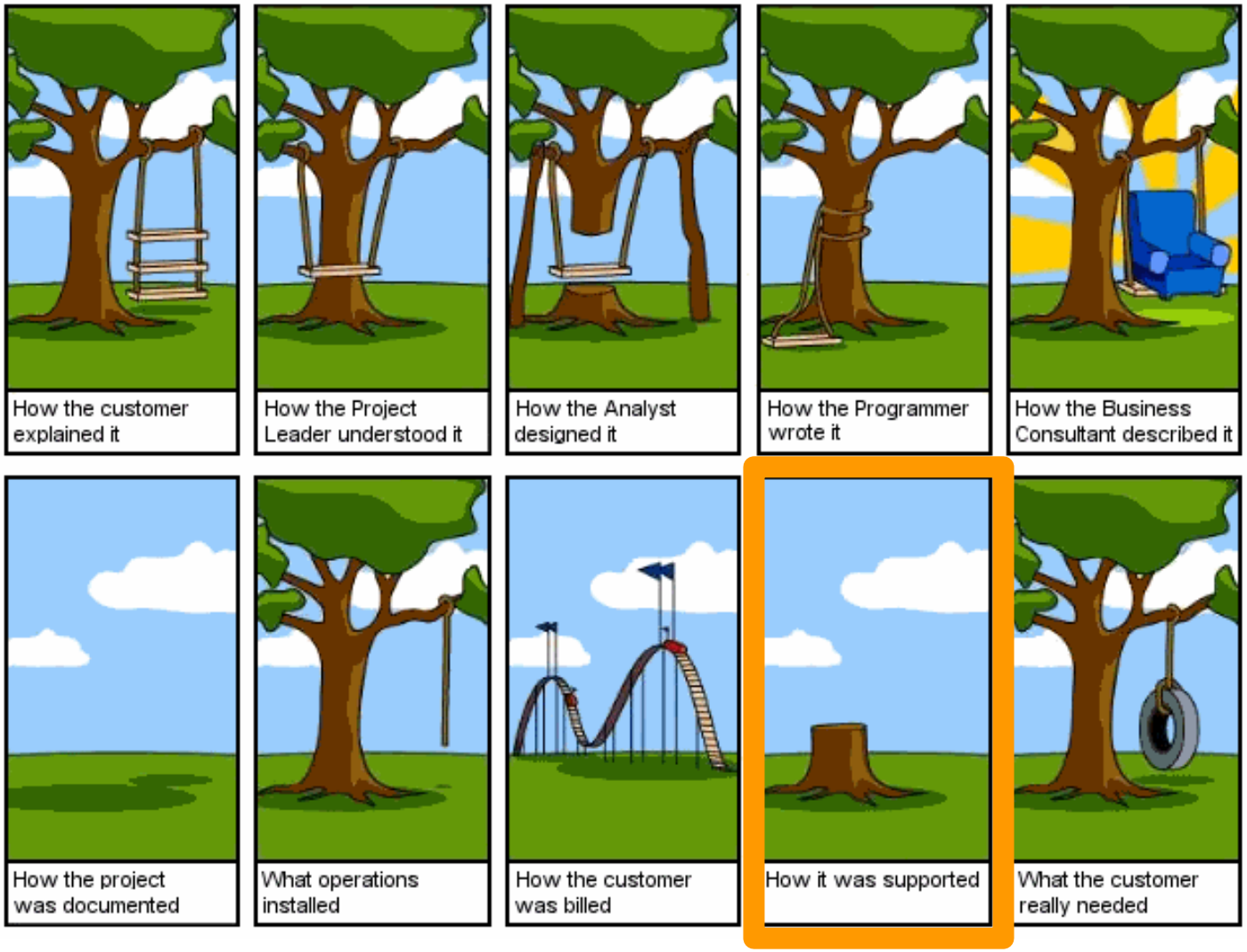
JE NE SUIS PAS EXPERT

Disclaimer

~~Basé sur une histoire vraie~~

Toute ressemblance avec un nom de produit, de projet, d'organisation ou personne existant ne serait que pure fortuite.

Il était une fois, le service Ni-Ni



Avait pour mission...

Et avec un mars, ça repart ?

Délai de restauration de service des Incidents			N° Indicateur	
Type Indicateur	KPI	**Unité de mesure**	%	
Fréquence de mesure	Mensuelle	**Moyen de mesure**	Outil de suivi	
Définition	Le délai de restauration de service d'un Incident sur une Application correspond au temps écoulé entre l'affectation à XX d' un ticket Incident et la restauration du service par une solution de contournement. Le délai court pendant la plage des horaires ouvrés de la TMA (24/7), lorsque le ticket est affecté à XX et hors statut « en attente ». A la clôture du ticket, le délai correspond à la période durant laquelle l'incident a été actif et affecté à XX jusqu'à la restauration du service et le fonctionnement de l'Application.			
Mode de Calcul	**Assiette de référence :** Ensemble des tickets incident affectés à XX ayant fait l'objet d'une restauration de service et clôturés dans le mois examiné. [(Nombre des éléments de l'assiette de référence résolus dans le respect des Niveaux de Service Requis) / (Nombre Total des éléments de l'assiette)]*100			
Priorité	**P1**	**P2**	**P3**	**P4**
Exigence service	**OR**			
Niveau Service Requis	**2 heures**	**4 heures**	**1 jour**	**Date planifiée**
Cible	**100%**	**100%**	**90%**	**90%**

Vive la virtualisation !

- Mise en place et maintient des environnements pour reproduction
  - VM & snapshots
- Reproduction des problèmes
  - Configuration personnalisée
  - Reproduction et analyse
- Escalade et validation des correctifs
  - Sélection du bon “snapshot”
  - “Même joueur joue encore”

Le problème d'Emile

On peut traiter 10 cases pour 1 client mais on ne peut pas traiter 100 clients euh .. 100 cases

- Multiplication des clients
  - Multiplication des versions
  - Multiplication des configurations/en
- Snapshot à gogo
  - Erreur de manip
  - Switch entre les différents intervenants

L'abus de snapshots est dangereux

- VM obèses (+300 Go)
- Difficile de démarrer plusieurs snapshots en même temps sur le même poste (comparaison)
- Qui ment ?
  - DEV = Client = Erreurs de manip !

Déploiement avec Docker

- n'importe quoi ?
  - les applications
  - les middlewares
  - les serveurs d'applications
  - les bases de données

- n'importe où ?
  - machine physique (archi off. Intel 64 bits), VM
  - Kernel Linux 3.8+
  - toutes les distro (binaires disponibles)
- par n'importe qui ?
  - même procédure
  - mêmes dépendances (libs, distro, etc.)

Docker: quelques concepts (survol)

- Container
- Image
- Engine
- Docker Hub
- Dockerfile

Docker: Container

- LXC (Linux Container)
- Isolation par namespaces (isolation des groupes de processus)
- Isolation par cgroups - control groups (isolation/limitation des ressources)

Docker : Image

- construite à partir d’un Dockerfile
- c’est la “classe”

Docker: Engine

- moteur open source écrit en Go
- commande CLI ⇒ requête JSON envoyée au démon docker
- crée des containers à partir des images

Docker Hub

- dépôt d’images
- même principe que Github
- Registry == “on premise”

Dockerfile

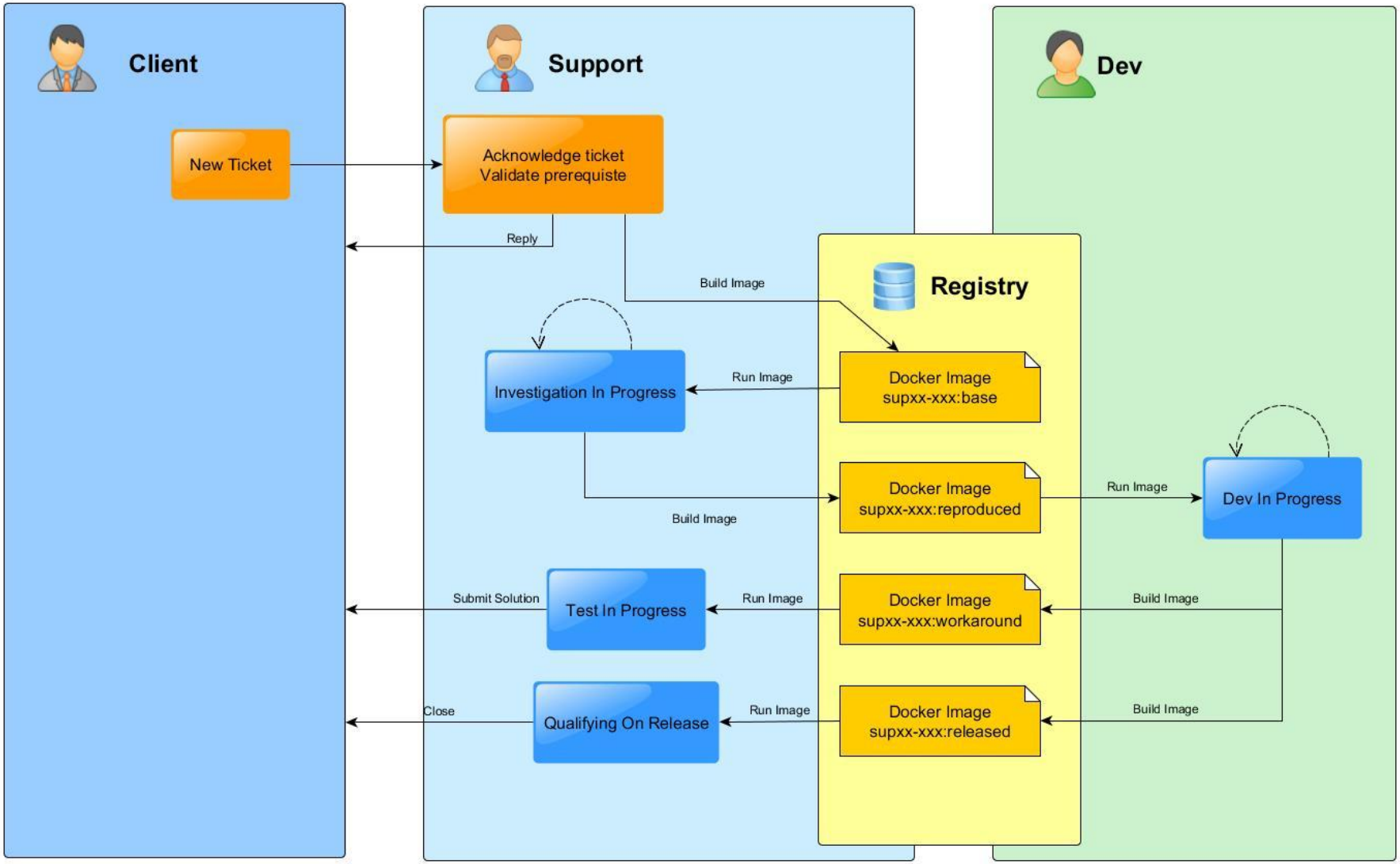
- fichier de description/instruction
- syntaxe simple à comprendre/lire (format texte)

```
FROM      ubuntu:latest
MAINTAINER Docker

RUN apt-key adv --keyserver hkp://p80.pool.sks-keyservers.net:80 --recv 7F0CEB10
RUN echo 'deb http://downloads-distro.mongodb.org/repo/ubuntu-upstart dist 10gen' | tee
/etc/apt/sources.list.d/10gen.list

RUN apt-get update && apt-get install -y mongodb-org
RUN mkdir -p /data/db
EXPOSE 27017
ENTRYPOINT ["/usr/bin/mongod"]
```

Revenons à nos moutons



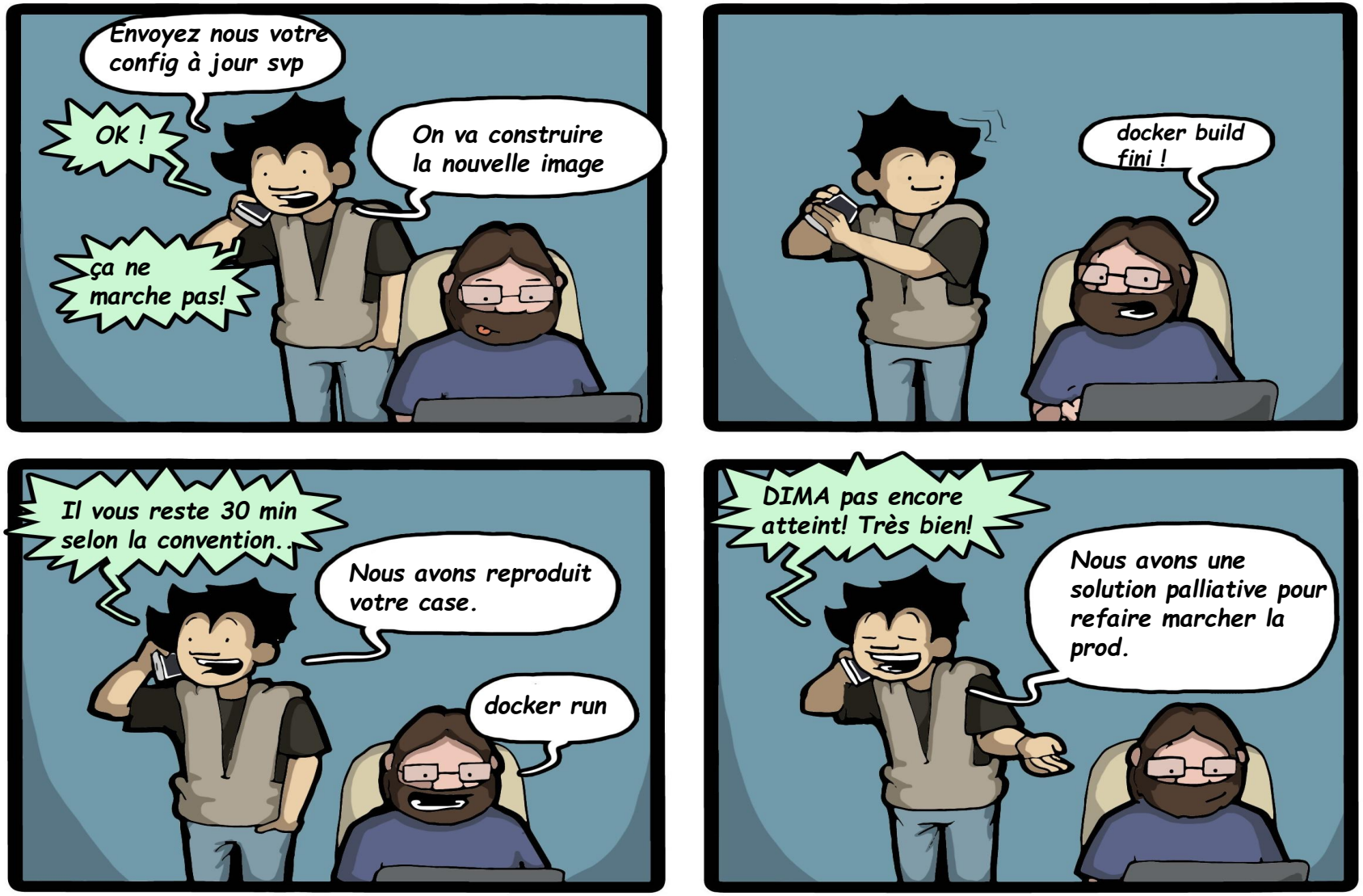
Et en plus

- même env de build que les DEV
  - AngularJS, bower, grunt, npm, etc sur Windows derrière un proxy corporate
- montage rapide d'une infra/cluster
  - exemple : plusieurs instances Mule ESB Enterprise + MMC (console de monitoring)

Réduction des délais

- Tout est dans l’image !
- Même procédure @Client et @DEV
  - Dockerfile traçable
  - simplification du suivi & de la maintenance des environnements
- Reproduction/Comparaison simplifiée

Et donc le SLA ?



Prochaine étape

- Validation des cases directement lors des builds
  - Utilisation des images des cases lors des CI
- Partage avec les clients
  - Dockerfile versionné
  - Construction automatique des Dockerfile
  - Build images & accès (registry)

Merci

DES QUESTIONS ?

--- J’ai pas de réponses ;)