

# TDO - SOIE02

---

## DOCKER

---

mercredi 4 octobre @ Prism

# Merci

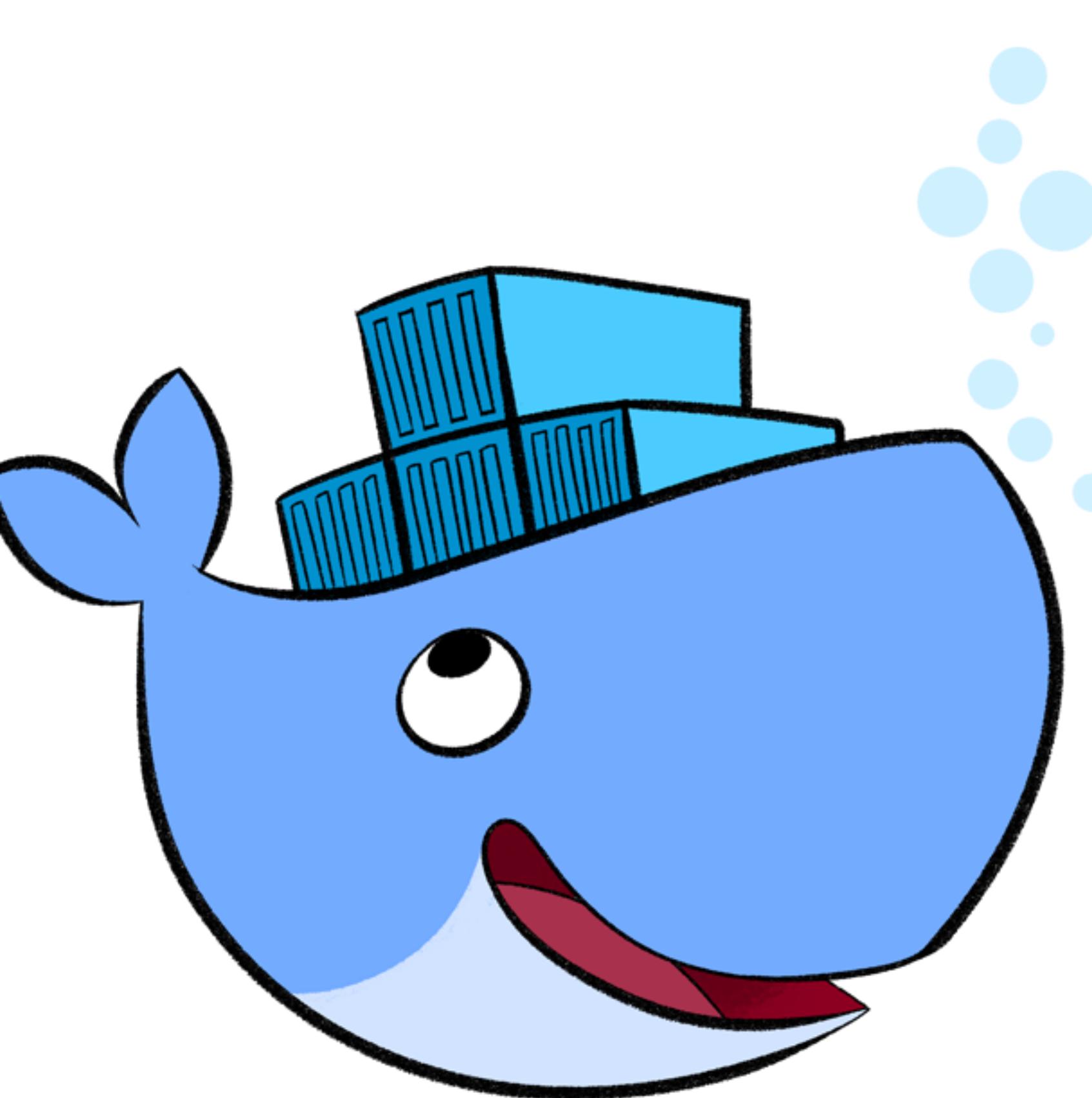
- au Prism de nous accueillir
- à la CCISM d'avoir créé le Prism
- à vous d'être venu

# Un mot sur cette présentation

- est un fichier Markdown
- versionné avec git
- transformé avec Deckset
- disponible sur github : source et PDF



Qui je suis...



# Sommaire

- C'est quoi ?
- Une petite démo
- REX
- Conclusion

# Qui connaît Docker ?

# Une sombre histoire de conteneur !!!

# Docker c'est :

- une équipe aimant la pizza qui parle français
- qui vont souvent à Bora, Moorea ou à la presqu'île
- Dockerfile, DockerHub, docker-compose.yml
- la valse des orchestrateurs (swarm, k8s and co)
- un tas de documentation sur le sujet

# Une petite démo

# Une petite démo

Ce qu'on va faire :

1. créer une application simple en java
2. mettre cette application dans un container
3. mettre les sources sur Github
4. configurer l'auto-build sur le DockerHub
5. déployer le container sur un cloud provider (DO)

# Une petite démo

## Créer une application simple en java (1/2)

- ne va faire qu'une seule chose : dire qu'elle va bien
- avec curl

```
mkdir demo  
cd demo  
curl https://start.spring.io/starter.tgz \  
  -d dependencies="web,actuator" \  
  -d name="demo" \  
  -d groupId="pf.devops" | tar -xzf -  
. /mvnw spring-boot:run
```

# Une petite démo

## Créer une application simple en java (2/2)

- ne va faire qu'une seule chose : dire qu'elle va bien
- avec spring-cli

```
/usr/local/bin/spring init -d="web,actuator" -g pf.devops -n demo demo  
cd demo  
.mvnw spring-boot:run
```

# Une petite démo

## Mettre cette application dans un container (1/3)

### → Création d'un Dockerfile (multi-stage)

```
FROM maven:3.5-jdk-8 AS appserver
WORKDIR /usr/src/app
COPY . .
RUN mvn package
```

```
FROM openjdk:jre-alpine
WORKDIR /app
COPY --from=appserver /usr/src/app/target/demo-0.0.1-SNAPSHOT.jar app.jar
ENTRYPOINT ["java", "-jar", "/app/app.jar"]
```

# Une petite démo

## Mettre cette application dans un container (2/3)

→ On lance la construction du container

```
docker build -t devops/demo .
```

# Une petite démo

## Mettre cette application dans un container (3/3)

→ On lance le container

```
docker run -p 8080:8080 -t devops/demo
```

→ On teste le service (avec httpie svp)

```
http :8080/health
```

# Une petite démo

On va héberger notre code sur Github

<https://github.com/tahitidevops/docker-demo>

```
git init
git add .
git commit -m 'Init project'
git remote add origin https://github.com/tahitidevops/docker-demo
git push -u origin master
```

# Une petite démo

## On configure Dockerhub

<https://hub.docker.com/u/tahitidevops>

# Une petite démo

On déploie le container sur Digital Ocean

<https://cloud.digitalocean.com/droplets>

```
doctl compute droplet create demo \
  --image docker-16-04 \
  --region sfo1 \
  --size 512mb \
  --ssh-keys 2b:b4:75:b5:14:47:76:93:05:49:31:27:54:81:1e:df
doctl compute droplet ls
doctl compute ssh demo
doctl compute droplet delete demo
```

# Une petite démo

Pour résumer...

- Créer un container, ce n'est pas sorcier
- Déployer un container non plus (il suffit d'avoir un dépôt de référence)

# Retour d'Expérience

# SIPF

# Retour d'EXpérience

## DOCKER & le SIPf :

- étudie Docker depuis 2013
- docker sur la plateforme Net.pf depuis 2016
- Dockercon 2017
- déploiement global via le SDI 2017

# Retour d'EXpérience

## DOCKER & les agents du SIPf :

- beaucoup plus simple avec les personnes habituées à la ligne de commande
- beaucoup plus compliqué avec les autres
- très compliqué avec les pro-silos !!!
- le CI/CD est votre allié pour rendre docker non visible

# Retour d'EXpérience

## DOCKER & les agents de l'Infrastructure :

- habitués aux développements des outils de gestion de configuration (Vagrant, Puppet, Chef & Ansible) donc docker... finger in the nose...
- tant que ça ne dépasse pas le poste de travail...

# Retour d'EXpérience

## DOCKER & la « réalité » de la production (1/3) :

- c'est plus compliqué que ça en a l'air
- la production, c'est de toute façons compliqué de base (Disponibilité, Intégrité, Confidentialité, Traçabilité)... la supervision, la centralisation des logs, les N plans au cas où...etc

# Retour d'EXpérience

## DOCKER & la « réalité » de la production (2/3) :

- doit-on tout mettre dans des containers ?
- quel orchestrateur ? quel architecture ?
- quelle application ? quid des délais ? quels sont les moyens et les ressources nécessaires ?
- Quid de la sécurité ? "On dit que" ?

# Retour d'EXpérience

## DOCKER & la « réalité » de la production (3/3) :

- “Voir grand mais commencer petit” dixit mon CDS
- dépend de votre application (micro-service ?)
- dépend de votre niveau de maturité avec l'éco-système Docker
- tout le reste, c'est comme d'hab... vraiment

# Conclusion

# Un mot sur l'association Tahiti DevOps

- Promouvoir la culture devops en Polynésie française
- Présentations en fonction des souhaits des adhérents
- <http://www.devops.pf>

