

Encapsuler une application R avec Docker

Les Rencontres R

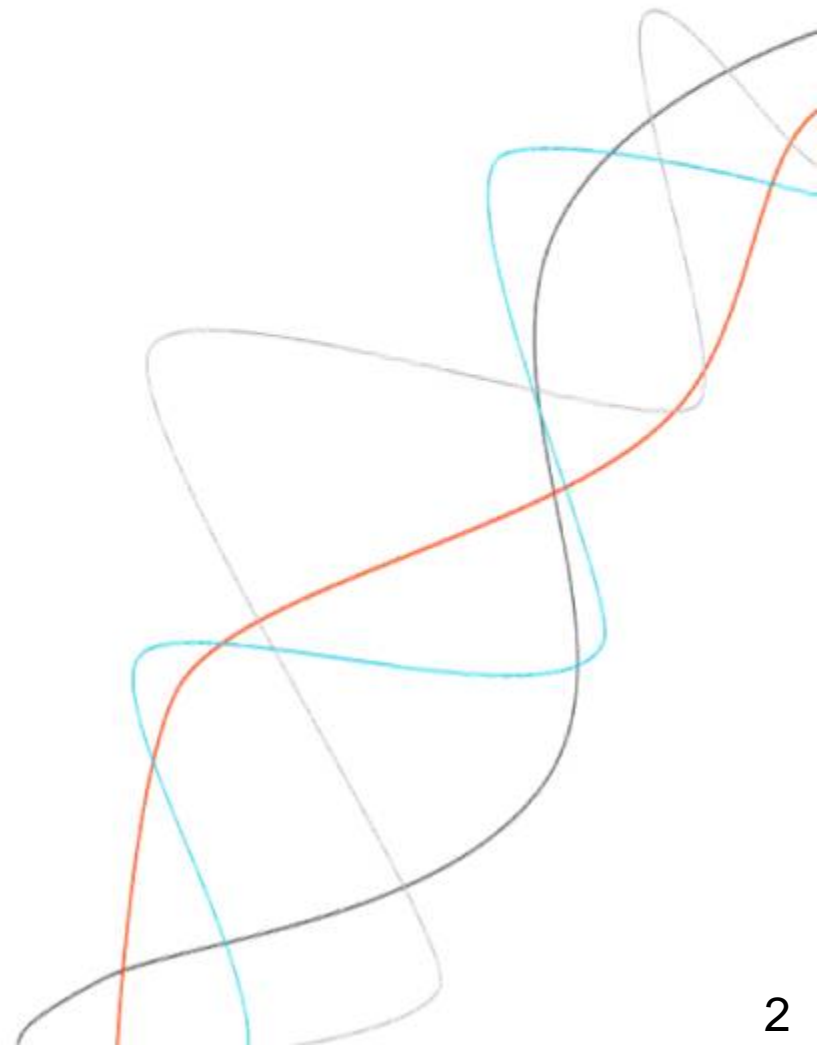
29 juin 2017

Vincent Guyader

vincent@thinkr.fr



L'EQUIPE



L'Equipe



Vincent Guyader

Codeur Fou, formateur et expert
logiciel R



Diane Beldame

Dompteuse de dragons données,
formatrice logiciel R



Romain François

Dactive – Rcpp Ninja



Colin Fay

Data Analyst, Formateur R, expert
Social Media



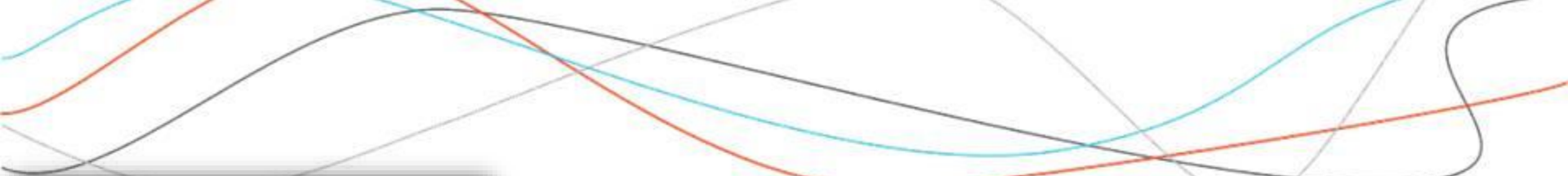
THINKR

https://twitter.com/thinkr_fr

<http://www.thinkr.fr>

LIVRER DU CODE R





fcuk

New Project...

Open Project...

Open Project in New S

Close Project



New Project

Create Project

New Directory
Start a project in a brand new working directory

Existing Directory
Associate a project with an existing directory

Version Control
Checkout a project from a version control system

New Project

Back

Project Type

Empty Project
Create a new project in an empty directory

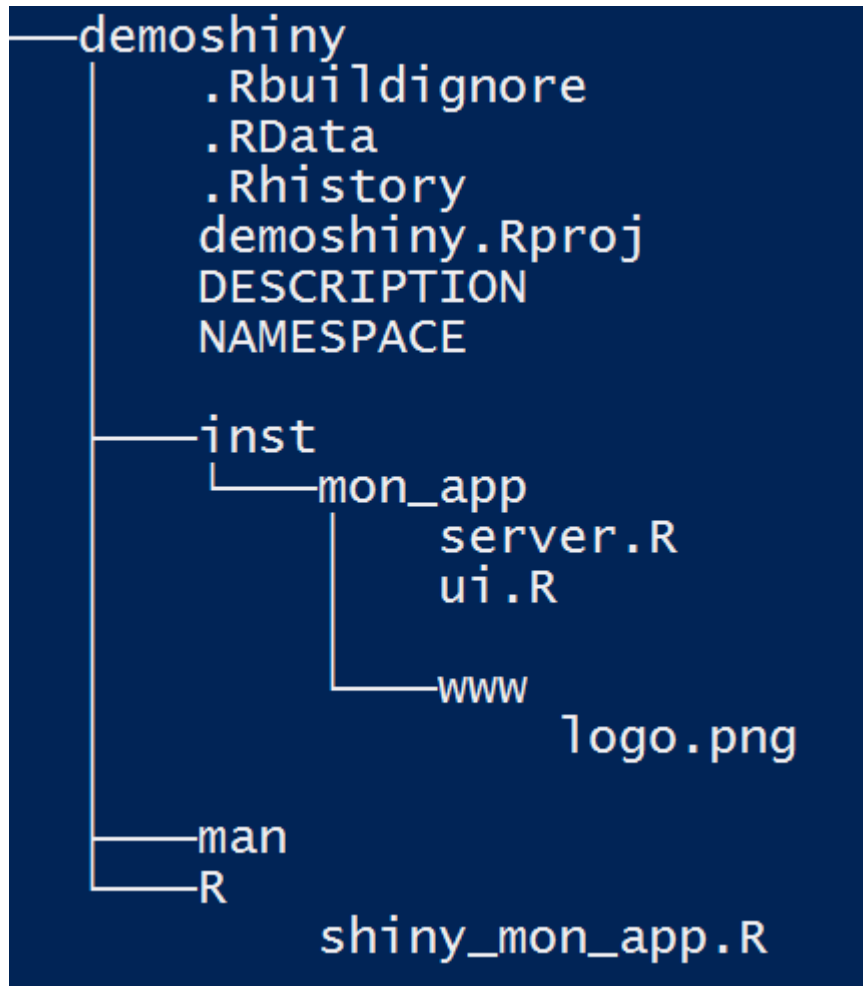
R Package
Create a new R package

Shiny Web Application
Create a new Shiny web application

ENCAPSULER UNE APPLICATION SHINY

Encapsuler une application Shiny dans un package

Exemple d'arborescence à mettre en place :



Les fichiers server.R et ui.R (et www) qui constituent l'application shiny à proprement parler

Le fichier .R qui contient, entre autres le code source la fonction qui lancera l'application

Coder une fonction qui lance votre application

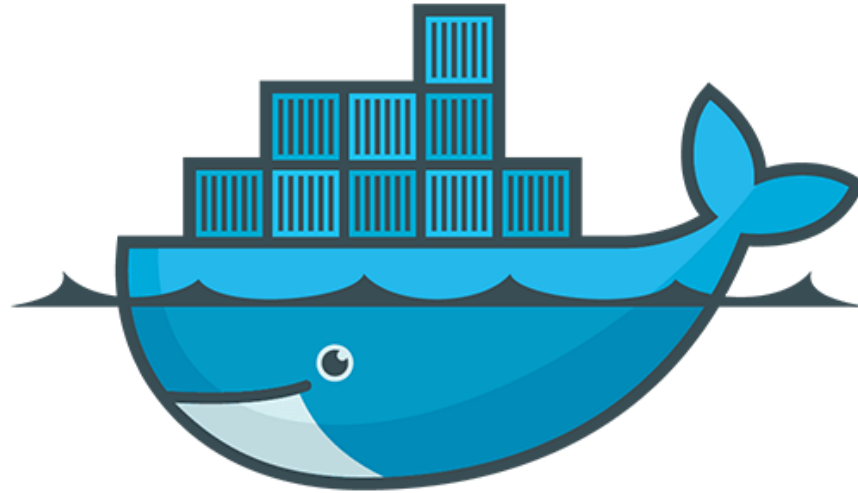
```
shiny_mon_app <- function() {  
  appDir <- system.file("mon_app", package = "demo  
shiny")  
  shiny::runApp(appDir, display.mode = "normal")  
}
```


Moby dock, votre nouvelle meilleure amie

DOCKER



C'est quoi Docker ?



docker

Docker est un logiciel libre récent (2013), plébiscité par sa « simplicité » et son efficacité. Il permet d'empaqueter une application et ses dépendances dans un conteneur isolé, qui pourra être exécuté sur n'importe quel ordinateur.



Pourquoi utiliser Docker (avec R) ?

- facile à mettre en œuvre
- simplifie/supprime les soucis de dépendance
- isole complètement un environnement
- facilite la portabilité de l'application
- la solution est réutilisable et amendable facilement

Pour bien comprendre Docker

Notions importantes :

- ***container*** : ce qui va faire tourner les applications
- ***images*** : environnement pré-conçus qui décrivent ce que devront faire les containers
- ***registry*** : dépôt qui regroupe les images de la communauté
- ***Dockerfile*** : fichier texte qui correspond à la « recette » d'une image (permet de construire une image)

Exemple d'utilisation de docker

Lancer une instance d'ubuntu :

```
docker run -it ubuntu
```

Lancer un Rstudio serveur sur le port 80 :

```
docker run -d -p 80:8787 rocker/hadleyverse
```

Lancer wordpress

```
docker run -p 8081:80 wordpress
```

Docker – autres commandes utiles

Gérer les containers

`docker ps`

-> liste les containers qui tournent sur la machine

`docker ps -a`

-> liste tous les containers, même ceux qui ne tournent pas

`docker stop <container-id>`

-> stoppe le container qui était en train de tourner (`docker kill <container-id>` en est la version brutale)

`docker start <container-id>`

-> relance le container qui était en train de tourner

`docker exec -it <container-id> bash`

-> ouvre un terminal dans un container en cours d'exécution

SHINY DANS DOCKER

Docker

Il faut créer sa propre image

-> Dockerfile

Un Dockerfile est une succession d'instructions, 1 ligne = 1 instruction.
Chaque instruction commence par une de ces balises :

FROM : précise l'image de base qui sera utilisée en point de départ

MAINTAINER : précise l'auteur de l'image

RUN : lance une instruction bash classique

COPY : copie un fichier de la machine vers l'image

EXPOSE : ouvre des ports sur la machine

CMD : indique la commande à exécuter juste après avoir lancé un container

Dockerfile

```
FROM thinkr/rfull

COPY demoshiny_*.tar.gz /demoshiny.tar.gz

RUN R -e "install.packages('demoshiny.tar.gz', repos =
NULL, type = 'source')"
```

```
COPY Rprofile.site /usr/local/lib/R/etc

EXPOSE 3838
CMD ["R", "-e demoshiny::shiny_mon_app()"]
```

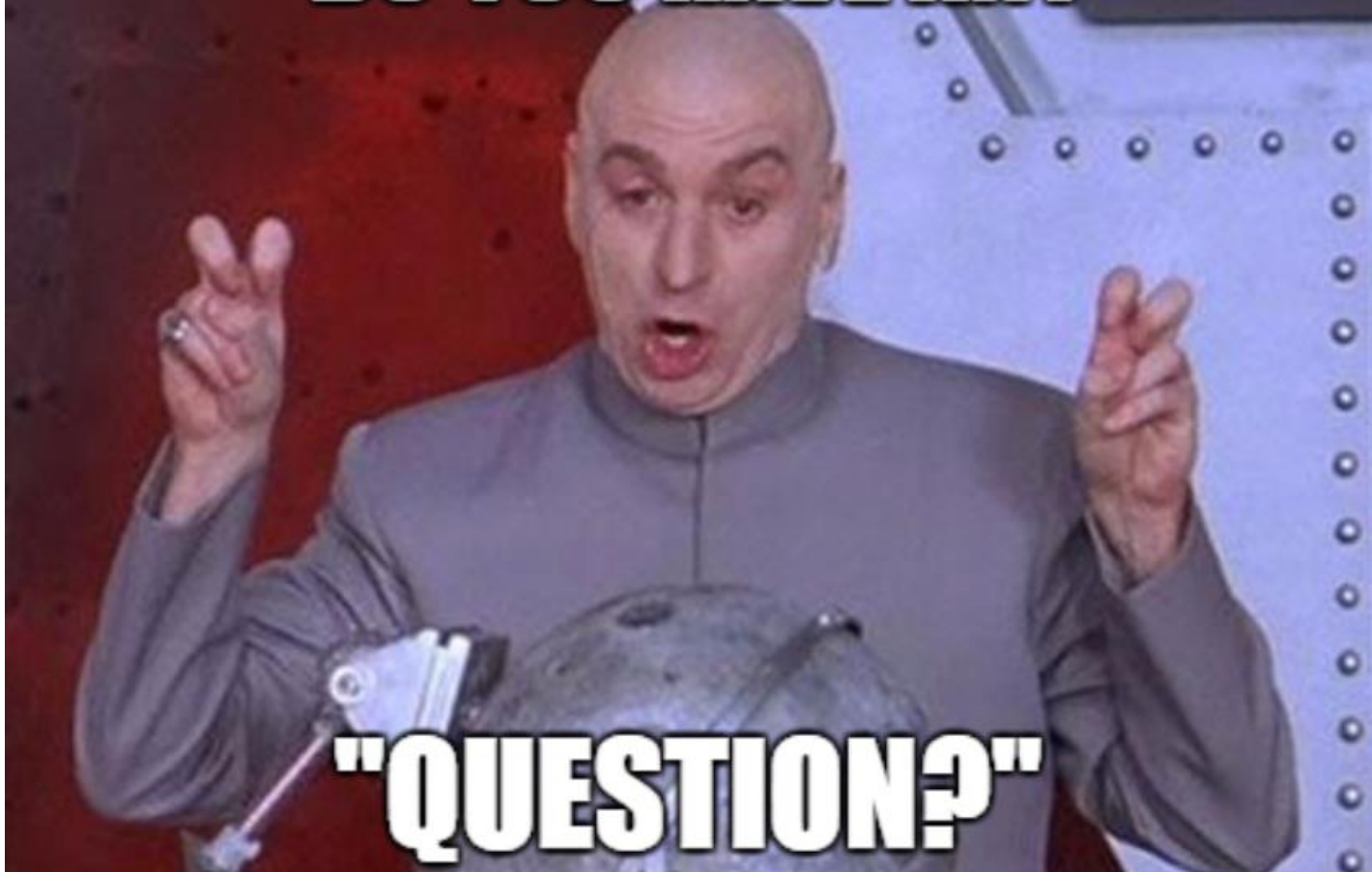
CREATION DE L'IMAGE & LANCEMENT DE L'APPLICATION

```
docker build -t demoshiny .
```

```
docker run -d -p 3838:3838 demoshiny
```

Puis <http://localhost:3838>

**THANKS FOR YOUR ATTENTION,
DO YOU HAVE ANY**



"QUESTION?"



Plus d'informations sur le blog

www.thinkr.fr