Construction et déploiement d'applications web basées sur R

Programme

1. Introduction : R et web
2. Présentation brève de plumber
3. Démo sur plumber
4. Notion de déploiement et dockerisation
5. Demo sur la dockerisation sous R utilisant Rapp

Alassane Samba, Chercheur à Orange Labs

Les Septièmes Rencontres R

Rennes, 05/07/2018

orange™

Introduction : R et web

- R né en 1993 comme langage d'analyse de données
- Aujourd'hui : émergence de plus en plus de packages permettant de programmer des applications web
- D'abord on a les packages permettant d'instancier des servers web : httpuv, Rserve, etc.
- Plusieurs packages s'appuient directement sur ces générateurs de server web, pour fournir des frameworks d'API sous R : plumber, opencpu, jug
- Le package shiny permet de construire facilement des interfaces web adaptées à la visualisation
- Le package htmltools et plusieurs autres qui en dépendent (plotly, visNetwork, rAmCharts, etc.) permettent de générer facilement des figures en HTML/Javascript
- Le package knitr permet de générer des rapports notamment au format HTML, en reprenant le principe du Markdown

Package plumber

- **Le package plumber [Jeff Allen et al., 2014], aujourd'hui intégré comme produit de la société Rstudio, est un framework facilitant la création d'API sous R**
- **Il permet la création d'une API sous R par simple décoration d'un script R avec des commentaires spéciaux**

- **https://www.rplumber.io**

- **→ DEMO**

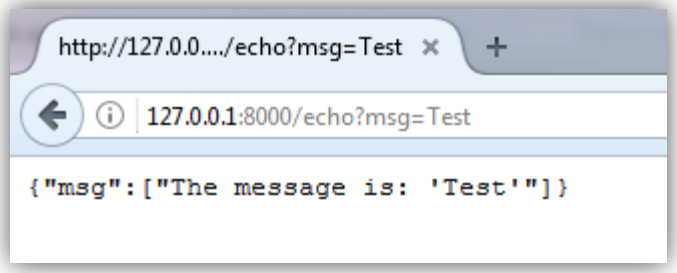
DEMO ‘Plumber ’ : script R à transformer en API

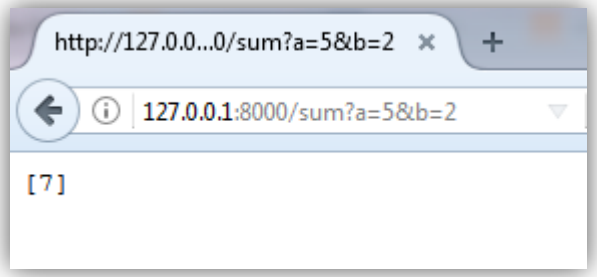
```r
# api.R

#* Echo back the input
#* @param msg The message to echo
#* @get /echo
function(msg=""){
  list(msg = paste0("The message is: '", msg, "'"))
}

#* Plot a histogram
#* @png
#* @get /plot
function(){
  rand <- rnorm(100)
  hist(rand)
}


#* @get /sum
function(a, b){
  as.numeric(a) + as.numeric(b)
}

```

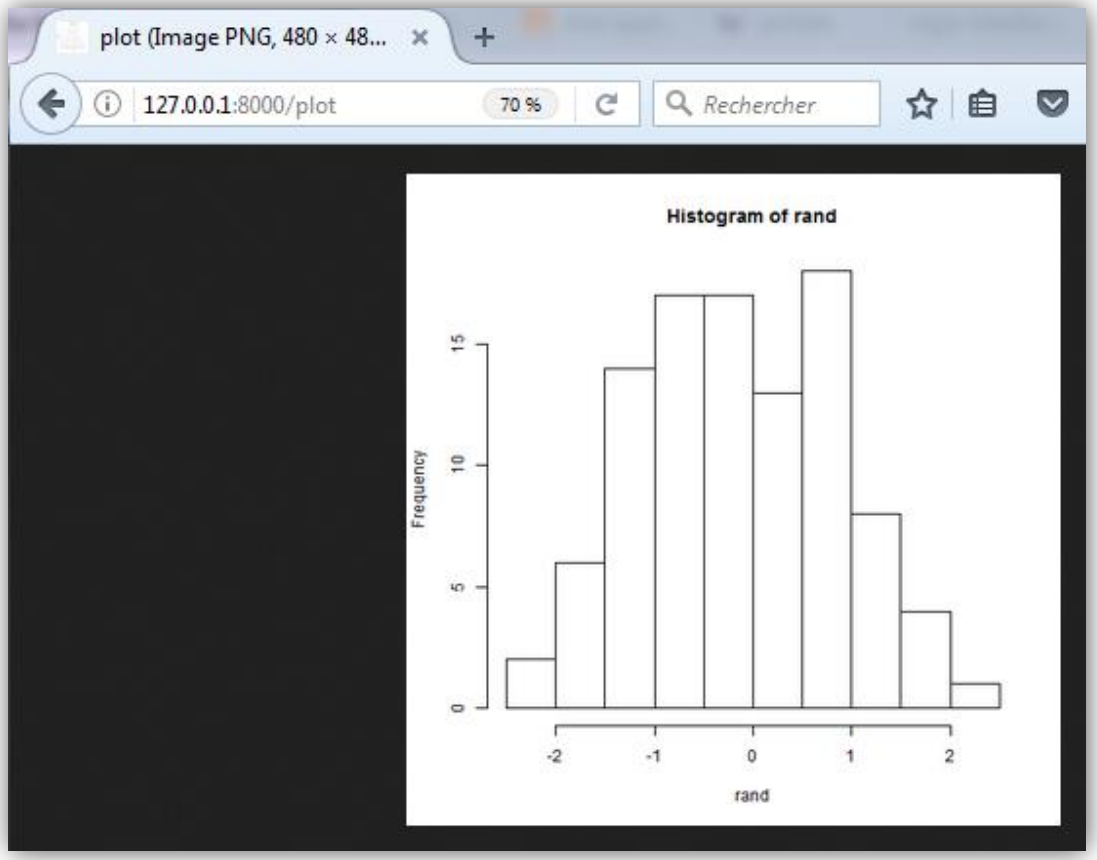
DEMO ‘Plumber ’ : lancement de l’API

```
> library(plumber)
> r <- plumb("app/api.R")
> r$run(port=8000)
Starting server to listen on port 8000
Running the swagger UI at http://127.0.0.1:8000/__swagger__/
```

DEMO ‘Plumber ’ : requête de l’API à travers un navigateur





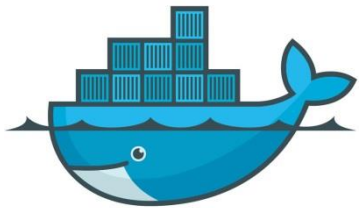


Déploiement des applications R dans des Clouds

- **Emergence de l'intelligence artificielle : besoin grandissant d'intégrer du Machine Learning aux applications web (analyse/prédiction/classification/… en temps réel)**
- **Enjeux: accessibilité, interopérabilité, robustesse, fiabilité, flexibilité**
- **→ Bénéficier du développement des technologies Cloud et les techniques issues du DevOps, notamment la containerisation**
- **→ Créer des interfaces pour utiliser sous R les outils Cloud et DevOps (git, Docker, CI/CD, interface Swagger pour les APIs, etc.)**

Containerisation avec Docker

- **Alternative légère à la virtualisation complète des machines qui consiste à encapsuler une application dans un conteneur avec son propre environnement d'exploitation**
- **Permet d'exécuter plusieurs processus et applications séparément les uns des autres afin d'optimiser l'utilisation d'une infrastructure Cloud**



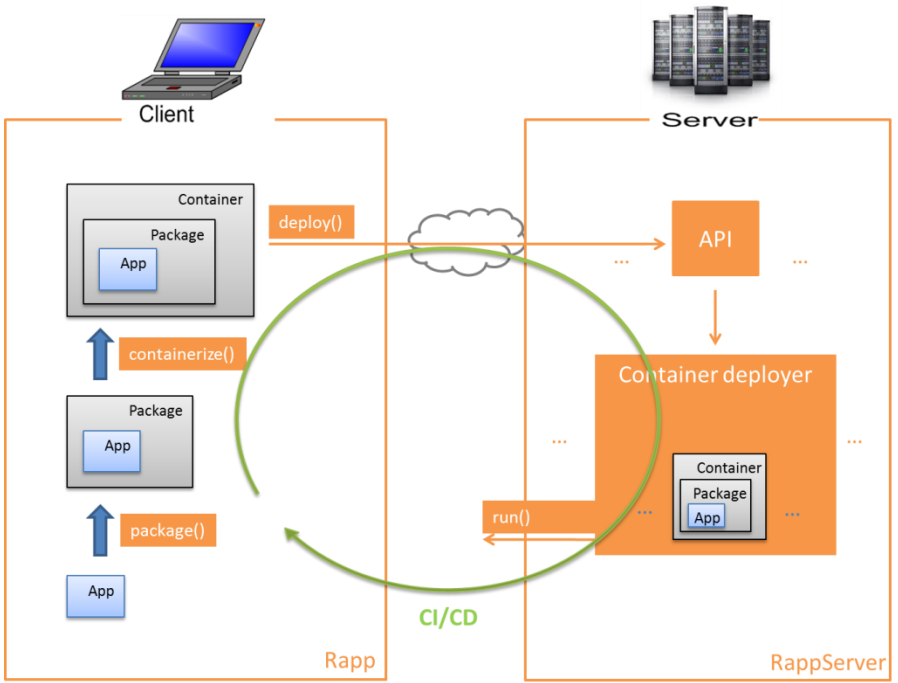
- **Docker est l'outil (open source) de containerisation le plus populaire aujourd'hui.**
  - Modèle de déploiement basé sur une 'image' : partage facile d'une application ou un ensemble de services, avec toutes leurs dépendances, entre plusieurs environnements

Dockeriser les applications shiny et les APIs plumber sous R avec Rapp

- **Dockeriser une application revient à créer un fichier Dockerfile permettant de construire entièrement une image Docker correspondant à l'application à déployer.**
- **Rapp & RappServer sont 2 packages en cours de développement facilitant respectivement**
  - **la dockerisation sous R**
  - **le déploiement dans un Cloud distant**

  des applications shiny et des APIs plumber, pour l'instant.

- **→ DEMO**

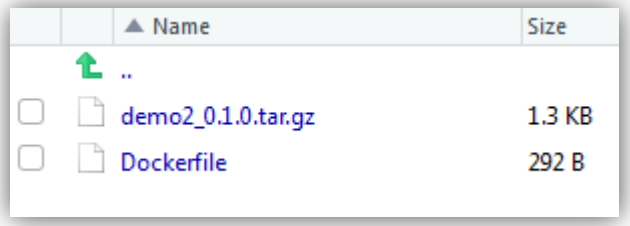


DEMO ‘Rapp ’ : Dockerisation de l’API

Packager-containeriser avec Rapp

```r
# Dockerisation
library(Rapp)
Rapp::package_and_containerize(appFolder = 'app/', type = 'plumber', newPackageName = 'demo2',
                               appMainFileName = 'api.R', portToExpose = '8008')
```

Dossier produit



Contenu du Dockerfile

```
1  FROM sambaala/rapps
2  RUN R -e 'if(!require("plumber")) install.packages("plumber", repos="https://cran.r-project.org/")'
3  COPY demo2_0.1.0.tar.gz demo2_0.1.0.tar.gz
4  RUN R -e 'install.packages("demo2_0.1.0.tar.gz", repos=NULL)'
5  EXPOSE 8008
6  CMD R -e 'demo2::run(host="0.0.0.0", port=8008)'
```

Merci

Alassane. Samba@orange.com

Github : https://github.com/sambaala
Twitter : https://twitter.com/sambalassan
Linkedin : https://www.linkedin.com/in/alassane-samba-33061744/

orange™