

# MA SCIENCE SUR LE WEB

## DÉVELOPPER DES APPLICATIONS WEB EN RECHERCHE À DESTINATION DE DIFFÉRENTS PUBLICS

Corentin M. Barbu

INRAE et Ecole Doctorale ABIES (réseau ADUM)

9 avril 2025

## OBJECTIFS

- ▶ Savoir développer une application web pour rendre accessible des résultats de recherche ou des outils d'aide à la décision
- ▶ Voir plus largement les outils de publications sur internet

## PROGRAMME

- ▶ présentations des possibilités de mise en ligne suivant la complexité du projet et le public visé
- ▶ application minimaliste fonctionnant en local
- ▶ mise en ligne de l'application minimaliste sur un serveur

Note: les exemples seront tous en R mais les mêmes outils sont souvent aussi disponibles sous Python

# 3 GRANDS TYPES DE MISES EN LIGNE

Ma science sur  
le web

Corentin Barbu

## SITE INTERNET (PLUS OU MOINS INTERACTIF)

- ▶ Une collection de documents fixes : Quarto (Rmarkdown) ...  
ex: <https://mocoriba.fr/>
- ▶ ... mais des graphiques peuvent être interactifs (plotly) ...  
ex: <https://plotly.com/r/getting-started/>
- ▶ ... la page peut même contenir de mini applications (shiny)  
ex: <https://quarto.org/docs/interactive/shiny/>

Site Web en  
Quarto/plotly

Application  
basique

Les programmes  
compagnons

Le déploiement

## UNE APPLICATION WEB

- ▶ Une interface qui communique avec un serveur ...  
... pour un contenu interactif. ex: <https://ccexplorer.eu>

## UNE API (PLUMBER)

- ▶ Permettre la consultation de bases de données, plus ou moins  
traitées : ex: Swagger MoCoRiBA:  
[https://mocoriba.fr/mocoribaAPI/\\_\\_docs\\_\\_/](https://mocoriba.fr/mocoribaAPI/__docs__/)

## Section 1

# SITE WEB EN QUARTO/PLOTLY

Note: Si Rmarkdown (.Rmd) existe encore, il est progressivement remplacé par Quarto (.qmd)

## COMMENT ÇA SE PRÉSENTE

- ▶ Quarto en ligne de commande
- ▶ Quarto dans R (python)
- ▶ aide de quarto

## AVANTAGES ET INCONVÉNIENTS

- ▶ qmd → pdf ou html
- ▶ Organisation générale
- ▶ tables des matières automatiques
- ▶ Recherche dans les billets de blog et navigation blog facilitées
- × les html générés sont très lourds

## Section 2

# APPLICATION BASIQUE

# HELLO WORLD EN SHINY

Ma science sur  
le web

Corentin Barbu

Site Web en  
Quarto/plotly

Application  
basique

Les programmes  
compagnons

Le déploiement

## Section 3

# LES PROGRAMMES COMPAGNONS



## LE PROTOCOLE/PROGRAMME UBIQUITAIRE: GIT

- ▶ facilite la sauvegarde
- ▶ permet la restauration
- ▶ permet le développement en parallèle de plusieurs “branches”

- ▶ Fichiers ouverts directement par R (rds/csv/etc.), data/
- ▶ Base clé-valeur : ex. REDIS
- ▶ Base de données relationnelle : ex: PostgreSQL

## Section 4

# LE DÉPLOIEMENT

## SOLUTIONS DE MISE EN LIGNE

- ▶ Solutions d'hébergement d'applications (ex: SK8 <https://sk8.inrae.fr/> ou Shinyapps.io)
- ▶ Hébergement cloud/"local" (ex: OVH)
  - ⇒ serveur en ligne (linux ?)
    - ▶ nom de domaine libre (ovh, inrae)
    - ▶ installation de shiny server
      - sudo apt install shiny-server
    - ▶ placer son application au bon endroit
    - ▶ proxy (ex:nginx)  
ex: redirection ccexplorer

## PASSAGE À L'ÉCHELLE

- ▶ hébergement d'application : payer
- ▶ serveur géré ex: docker swarm

# PASSAGE À L'ÉCHELLE AVEC DOCKER ET TRAEFIK

Ma science sur  
le web

Corentin Barbu

Site Web en  
Quarto/plotly

Application  
basique

Les programmes  
compagnons

**Le déploiement**

Installation et mise à jour

- ▶ docker
  - ▶ permet la conteneurisation : “mini machines virtuelles”
  - ▶ un conteneur est de tailles beaucoup plus faible qu'une machine virtuelle
  - ▶ installations dans le conteneur indépendantes du système
  - ▶ déploiement ultra-rapide sur de nouveaux serveurs
- ▶ docker swarm
  - ▶ Permet la gestion de conteneurs multiples
- ▶ traefik
  - ▶ Permet de maintenir un utilisateur sur un même conteneur si besoin
  - ▶ Un conteneur additionnel par lequel passent toutes les requêtes

- ▶ “purification” des chaînes de caractères entrées  
ex: `entreePropre<-gsub('[^a-zA-Z0-9]','',entree)`
- ▶ Voir plus généralement les recommandations OPSWAT  
<https://www.opswat.com>
- ▶ authentification/gestion de droits par token

## LES CLASSIQUES

- ▶ pare-feu : ufw (linux) + généralement fourni par hébergeur cloud en amont
- ▶ https (letsencrypt/certbot)
- ▶ Antivirus (inrae: <https://www.withsecure.com/>)
- ▶ fail2ban : surveillance de logs
- ▶ authentification (Basic Auth)

## Subsection 1

# INSTALLATION ET MISE À JOUR



- ▶ Graphique :
- ▶ en ligne de commande linux : `rsync`  
`rsync -avz repSource/* user@server:repDest/`

<https://github.com/cbarbu/WebMyScience>

