



**HAL**  
open science

# Shiny-K8s, a toolkit to easily and reproducibly deploy your (R)Shiny app thanks to docker and Kubernetes (without mastering them!)

Bryan Brancotte, Elodie Chapeaublanc

## ► To cite this version:

Bryan Brancotte, Elodie Chapeaublanc. Shiny-K8s, a toolkit to easily and reproducibly deploy your (R)Shiny app thanks to docker and Kubernetes (without mastering them!). JOBIM 2024, Jun 2024, Toulouse (FRANCE), France. . pasteur-04643557

**HAL Id: pasteur-04643557**

**<https://pasteur.hal.science/pasteur-04643557v1>**

Submitted on 10 Jul 2024

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial - NoDerivatives 4.0 International License

# Shiny-K8s, a toolkit to easily and reproducibly deploy your (R)Shiny app thanks to docker and Kubernetes (without mastering them!)

Bryan Brancotte

Elodie Chapeaublanc

Bioinformatics and Biostatistics Hub, Institut Pasteur, Université de Paris, Paris, F-75015, France



Keywords : Shiny, Kubernetes, Reproducibility, DevOps, Docker

At the bioinformatics hub of Institut Pasteur - Paris, Our Shiny-K8s toolkit simplifies the deployment of R Shiny [1] and shiny-python apps [2]. It uses Docker [3], Kubernetes [4], Helm [5], and GitLab-CI/CD [6] for automated deployment. It promotes software development best practice with GitLab source hosting and new instance automatically deployed for any new branch. Features include auto-scaling, updates without downtime, persistent storage, and private project content. The toolkit provides step-by-step documentation, and is available under an MIT license. Multiple projects at the Institut Pasteur are already using the toolkit.

## Docker

Container technologies like Docker enable you to encapsulate your application, its dependencies, and the operating system within an image. Our toolkit proposes pre-built images featuring different versions of R/Python, Ubuntu, and a configured ShinyServer. On top of these "images" your app and its dependencies are added.

## Helm

How your application will run in kubernetes cluster is described by multiple templates. Helm allows to circumvent the configuration of these templates into one single small configuration file. Using true/false you unleash no-down-time, private registry, usage-based automated scaling of your app. Resources allocation (core, ram, storage) is also centralized in this file.

## Kubernetes

Kubernetes functions like a cloud, where containers replace VMs. Once your application is defined by templates, Kubernetes main task is to continuously adjust its state to align with your requirements: if a container crashes, it starts a new one; more storage asked, it is applied.

## Gitlab CI/CD

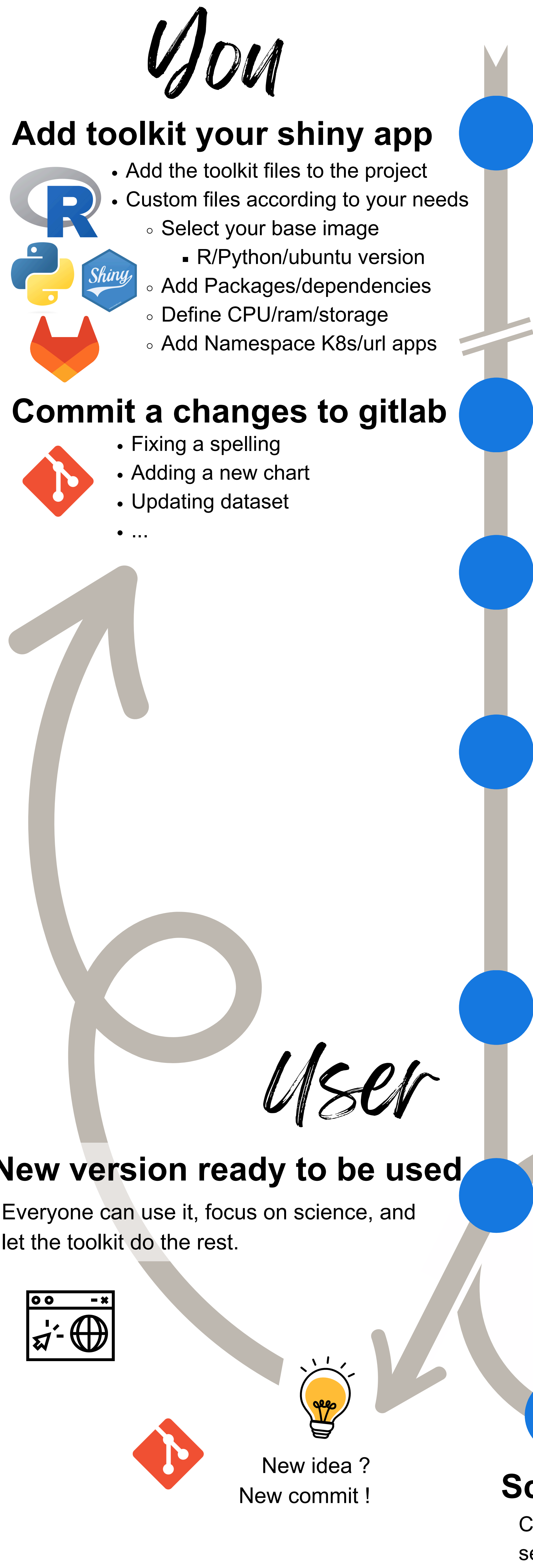
Continuous Integration and Deployment consists in having a workflow which test your app, build it, deploy it, package doc, publish it, ... CI/CD is triggered on any event such as commit. In the Shiny-K8S toolkit, we implemented a workflow where we build the image containing your app along with its dependencies, then use the template packaged by helm to describe how your app should be deployed, and actually transmit it to kubernetes. In this CI/CD workflow manually triggered task allows you to end data to your app in kubernetes

Doc

Source

Example

## The toolkit



### Build an image with your app

... this take the base image you selected it, and build a new docker image with your app. The image is pushed to a registry, ready to be used by k8s

### Configure the application in the k8s clusters

Indicate to kubernetes how to run your app :

- where is the image with you app
- what is the url to serve your app
- wether you need storage
- autoscaling
- no down time

### Stop previous version and start new

Kubernetes do it automatically once we have specified information from the previous step with the most important one : where is the image with your app

### Restart if it crashed

By a bug in your code, an OutOfMemory ? Kubernetes sees it and restart your app

### Scale up/down to fit usage

CPU/RAM to high ? If you enabled this option, a new server is started to face the surge of traffic

### New version ready to be used

Everyone can use it, focus on science, and let the toolkit do the rest.

User

New idea ?  
New commit !

### Facilitated and encouraged by the toolkit

- Your app is more **reproducible** : every changes are tracked, you can redeploy a previous commit at anytime.
- Using **git branches** is encouraged : when a new branch is created, a new instance of your app starts serving this version of your app, with no additional task to do, just git checkout -b my-next-feature
- **"I don't want my data in git"** Adding your data in the source repository is not always the best choice (private data, too big, regular new version, ...). You can ask the toolkit to fetch data from any given url.

### Plan B No Kubernetes?

You can still use the docker compose made for production settings, it pulls the image from the gitlab registry, and serve your app. Just ask your sysadmin a virtual machine, and an url routed to the VM.



1. Chang W, Cheng J, Allaire J, Sievert C, Schloerke B, Xie Y, Allen J, McPherson J, Dipert A, Borges B (2024). shiny: Web Application Framework for R. R package version 1.8.1.9000. <https://github.com/rstudio/shiny>, <https://shiny.posit.co/>.  
 2. Shiny for Python. Documentation available from <https://shiny.posit.co/py/>.  
 3. Merkel D. Docker: lightweight Linux containers for consistent development and deployment. Linux journal. 2014;2014(239):2.  
 4. Kubernetes is an open-source container orchestration engine for automating deployment, scaling, and management of containerized applications. The open-source project is hosted by the Cloud Native Computing Foundation <https://kubernetes.io/>.  
 5. Helm, the package manager for kubernetes. Documentation available from <https://helm.sh/>.  
 6. Gitlab CI, Documentation available from <https://docs.gitlab.com/ee/ci/index.html>