Streamlining R package development with GitHub Actions Workflows

useR! 2024, Salzburg

Daphné Grasselly - Senior Data Scientist, Roche
Franciszek Walkowiak - Senior IT Professional, Roche
Paweł Rucki, Principal Data Scientist, Roche

8th of July 2024

Speakers

Who we are?

Paweł Rucki
Principal Data Scientist, Roche

Daphné Grasselly
Senior Data Scientist, Roche

Franciszek Walkowiak
Senior IT Professional, Roche

TITLE	LENGTH (approx.)	TIME (approx.)
Introduction to CICD and GHA	25 min	9:25
Setup test environment	5 min	9:30
Exercise 1: “Hello World” from GHA	10 min	9:40
Exercise 2: R CMD CHECK workflow	40 min	10:20
Break	15 min	10:35
Exercise 3: Triggers	15 min	10:50
Exercise 4: Reusable workflows	20 min	11:10
Codespaces and Docker images introduction	15 min	11:25
Break	15 min	11:40
Exercise 5 (Bonus) : Play with Codespaces	15 min	11:55
Exercise 6 : Use docker images in workflows	10 min	12:05
Q&A	25 min	12:30

Sticky notes color code

If you're blocked during the exercices, or well advanced let us know!

🟥

I am blocked

🟩

Exercise finished / I can help other people

👍

Prerequisites for practical exercises

- If you'd like to follow along, you will need a GitHub account.
- We will modify files on GitHub using the Web IDE, so no further configuration is required.
- Optional for advanced users: if you prefer to edit and commit the files locally, you will need `git` installed on your computer and configured to have write access to GitHub (e.g. SSH key) (configuring this is out of scope of this workshop).

You can download these slides here https://sched.co/1c8yl :

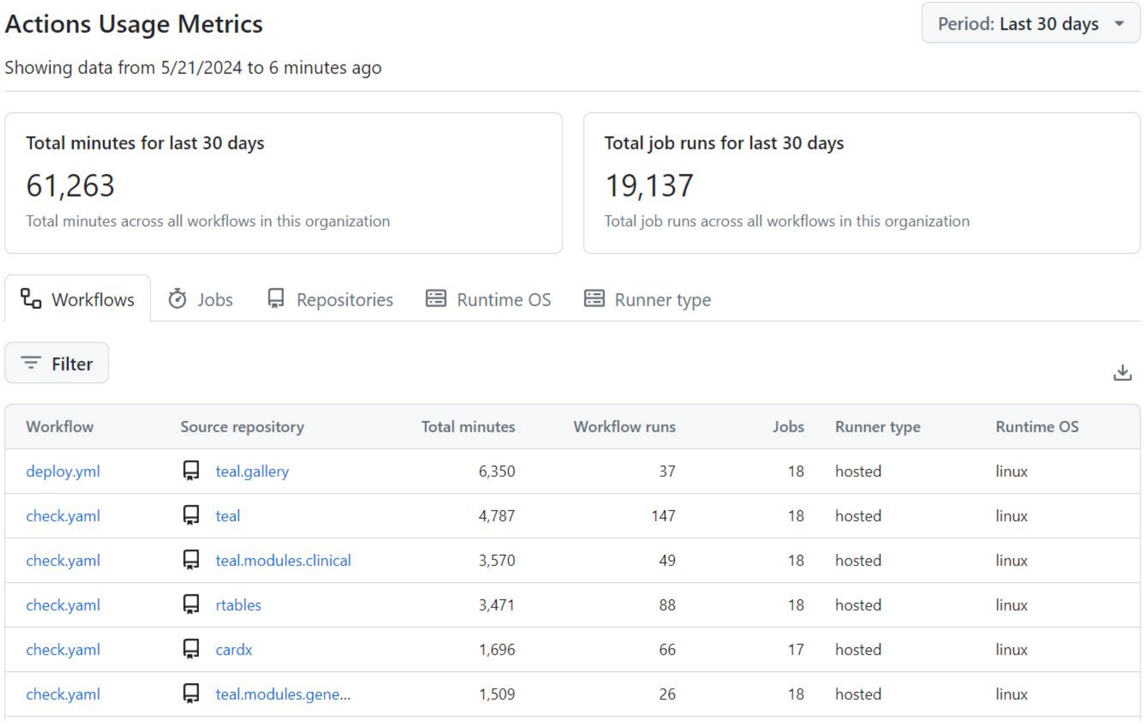
Streamlining R Package Development With Github Actions Workflows PDF

R workflow + deployment + production, Tutorial

https://sched.co/1c8yl

We are experienced with GitHub Actions and workflows 😊

There's quite a lot happening in **insightsengineering** GitHub organization:



Actions Usage Metrics

Period: Last 30 days

Showing data from 5/21/2024 to 6 minutes ago

Total minutes for last 30 days

61,263

Total minutes across all workflows in this organization

Total job runs for last 30 days

19,137

Total job runs across all workflows in this organization

Workflows | Jobs | Repositories | Runtime OS | Runner type

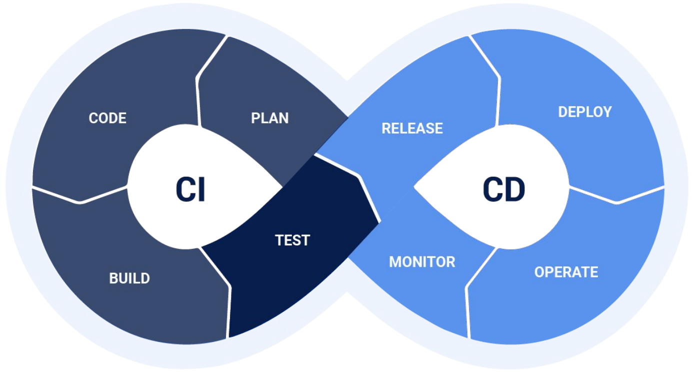
Filter

Workflow	Source repository	Total minutes	Workflow runs	Jobs	Runner type	Runtime OS
deploy.yml	teal.gallery	6,350	37	18	hosted	linux
check.yaml	teal	4,787	147	18	hosted	linux
check.yaml	teal.modules.clinical	3,570	49	18	hosted	linux
check.yaml	rtables	3,471	88	18	hosted	linux
check.yaml	cardx	1,696	66	17	hosted	linux
check.yaml	teal.modules.gene...	1,509	26	18	hosted	linux

Introduction to CI/CD and GitHub workflows

What is CI/CD?

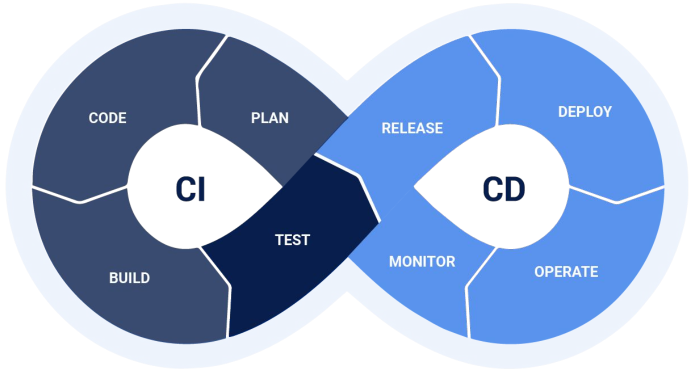
- **CI** : Continuous Integration
- **CD** : Continuous Deployment



- **Continuous Integration** is the practice of frequently integrating code changes into a shared repository, automatically verifying them through automated builds and tests.
- **Continuous Deployment** is the automated process of deploying those changes into production after passing CI checks.

What is CI/CD?

- **CI** : Continuous Integration
- **CD** : Continuous Deployment

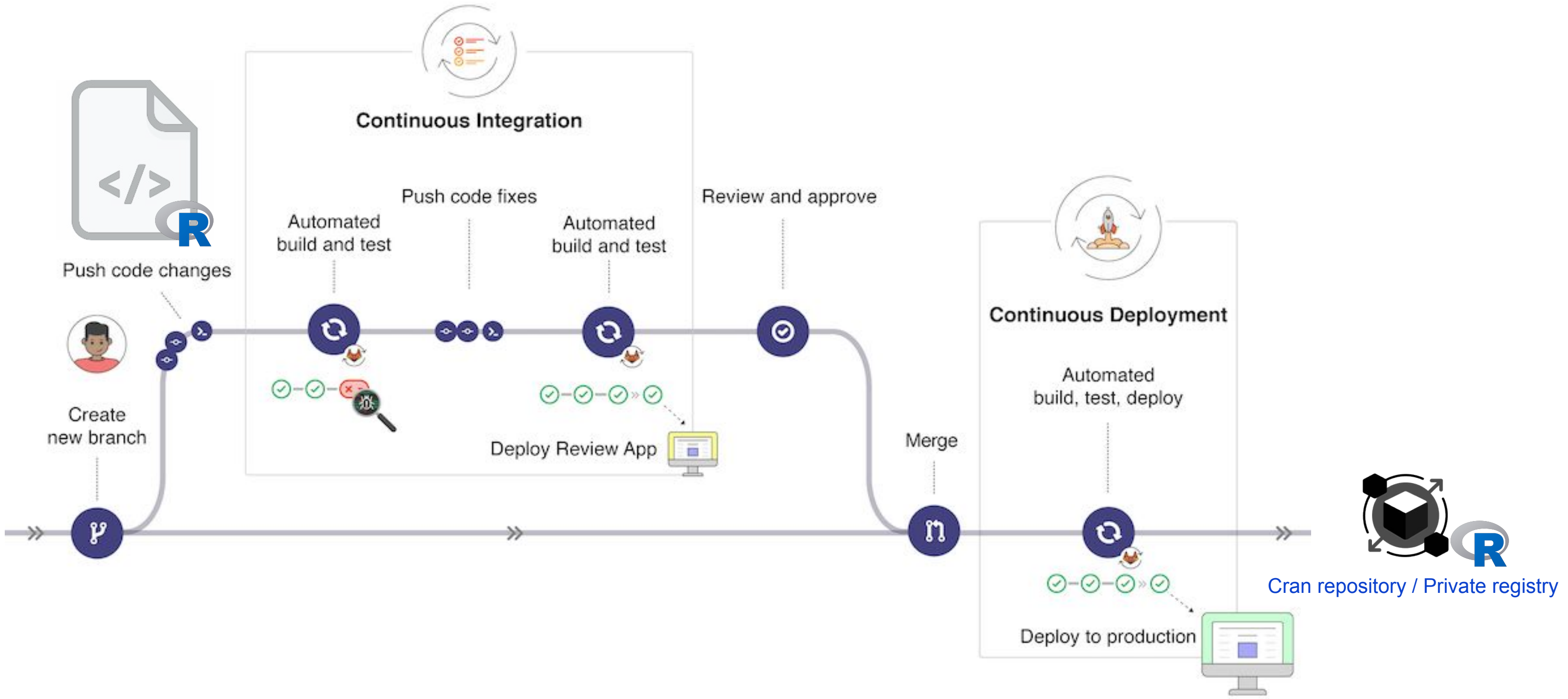


Why starting with CI/CD when working on a R package?

- Ensure delivering good code quality and maintainability
- Ensure to never bring regression when adding a new feature
- Reduces the time and resources required for manual deployment
- Globally enforcing rigor and good practices for package development

Classical workflow

Workflow schema with push and merge events



Usual steps

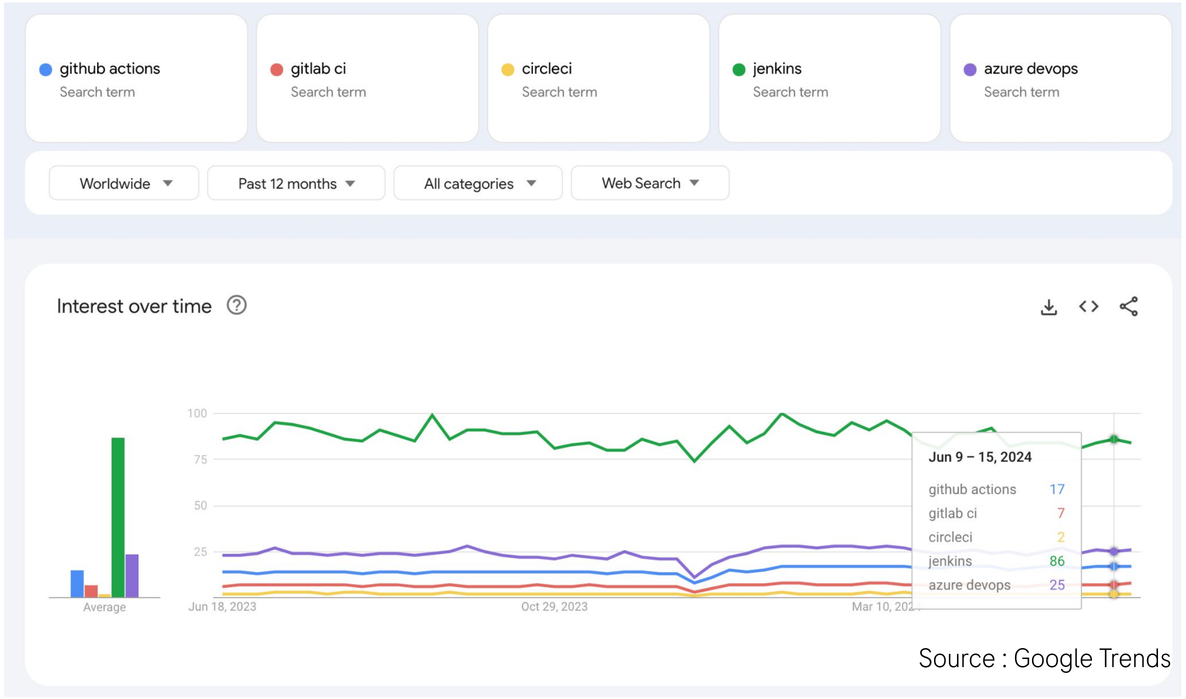
Usual steps for R package development

- **Code pushed to default branch, or code pushed to pull request branch:**
  - Code Style
  - Spelling
  - Lint
  - R CMD CHECK
  - Code coverage
  - Roxygen (code documentation)
  - Dependencies scan
  - Building user guide doc from .Rmd files with pkgdown

- **New release (git tag) created:**
  - Build a package, and attach it to GitHub release.
  - Publish package documentation to GitHub Pages.
- We have the insightsengineering/r.pkg.template repository with built-in GitHub workflows. You will find all the mentioned examples and even more!

Products to design CI/CD pipelines

Most used tools



Jun 9 – 15, 2024

github actions	17
gitlab ci	7
circleci	2
jenkins	86
azure devops	25

Source : Google Trends

For this workshop, we will focus on Github Actions workflows

Terminology: GitHub Workflows & GitHub Actions

Github Workflows vs Github Actions

GitHub workflow: set of **jobs** to run for a particular event occurs in the repository.

- Many workflows can run in parallel for the event.
- Usually some automated process (building, testing, deploying software etc.)
- **Jobs** consist of:
  - **steps**
  - Or other workflows imported from another repository.
- **Jobs** can be independent or dependent on each other.
- Workflows are stored in `.github/workflows` directory in the git repository.

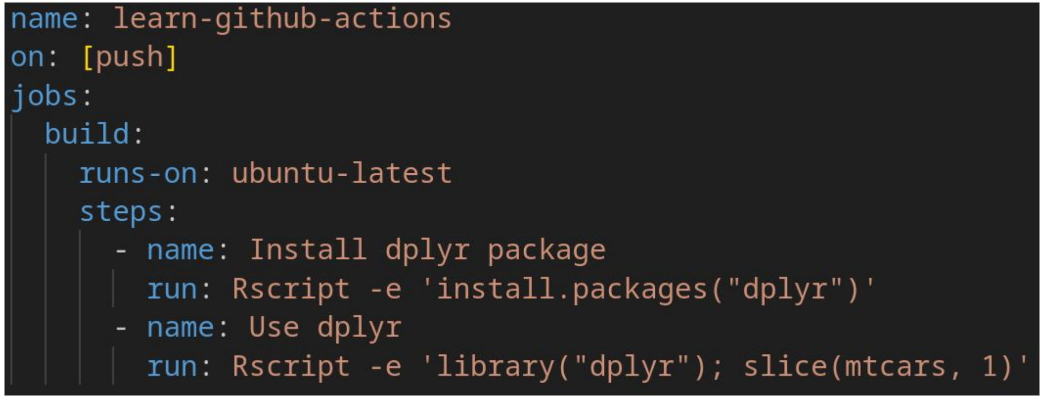
GitHub Action: reusable piece of code with parameters (inputs), expected to be reused as a **step** in a workflow.

- Typically each GitHub Action is developed in its own repository and can be imported from that repository.
- There's also GitHub Actions Marketplace.

GitHub Workflows

How to design a first workflow from scratch

- **Workflow structure (YAML syntax)**
  - **name:** Global name of the workflow
  - **on:** Event that will trigger the workflow (see list of possible events)
  - **jobs:** Job configuration (custom job, or calling a **reusable workflow**)
  - **runs-on:** GitHub runner (where all the jobs/steps will run) - it’s possible to choose between ubuntu/windows/mac-os (see more details here)
  - **Steps (each job in the workflow usually contains several steps)**
    - **uses:** To reuse existing **GitHub Action**
    - **run:** To run custom commands

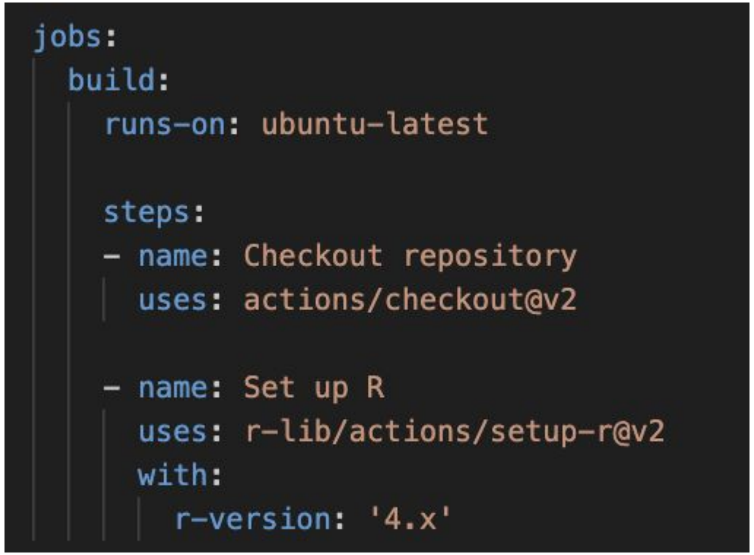


```yaml
name: learn-github-actions
on: [push]
jobs:
  build:
    runs-on: ubuntu-latest
    steps:
      - name: Install dplyr package
        run: Rscript -e 'install.packages("dplyr")'
      - name: Use dplyr
        run: Rscript -e 'library("dplyr"); slice(mtcars, 1)'
```

GitHub Workflows

How to design a first workflow from scratch

- **Your workflows will always need to contain these 2 first steps:**
  - **Checkout repository** step: fetches the repository's contents, enabling subsequent workflow steps to access and work with the latest version of the codebase.
  - **R environment setup** step: it will set up R for all the steps of the job.



```yaml
jobs:
  build:
    runs-on: ubuntu-latest

    steps:
    - name: Checkout repository
      uses: actions/checkout@v2

    - name: Set up R
      uses: r-lib/actions/setup-r@v2
      with:
        r-version: '4.x'
```

→ https://github.com/actions/checkout

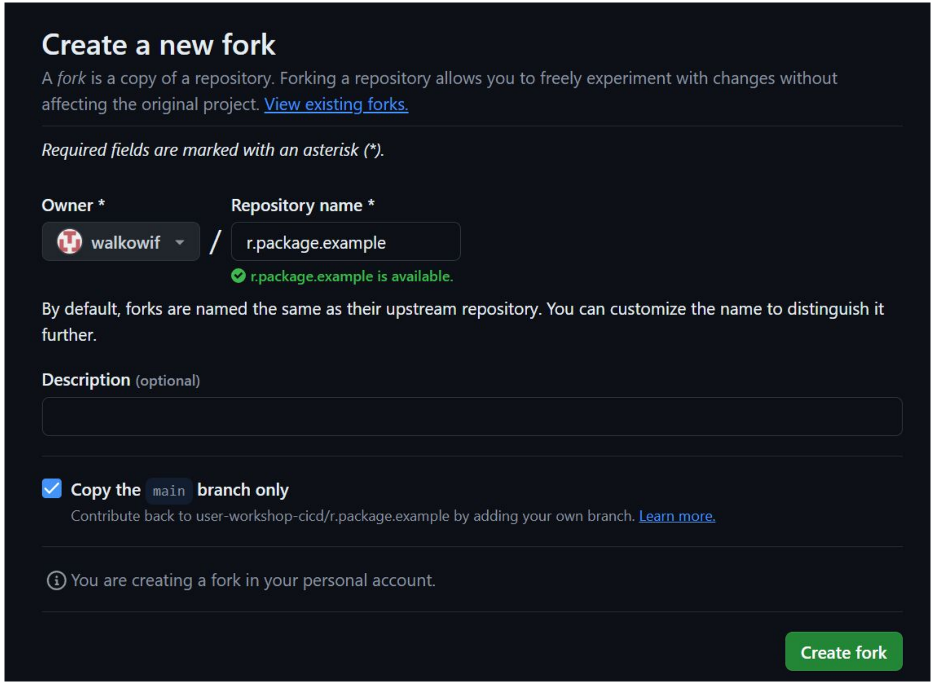
→ https://github.com/r-lib/actions/tree/v2-branch/setup-r

Exercises setup (create repository fork)

Roche

Exercises Setup

1. Log in to GitHub.
2. Go to https://github.com/user-workshop-cicd/r.package.example
3. Fork the repository (Fork → Create a new fork → Owner: yourself, ✔️copy main branch only)



Create a new fork

A *fork* is a copy of a repository. Forking a repository allows you to freely experiment with changes without affecting the original project. View existing forks.

Required fields are marked with an asterisk (*).

Owner * walkowif / **Repository name *** r.package.example

r.package.example is available.

By default, forks are named the same as their upstream repository. You can customize the name to distinguish it further.

Description (optional)

☑ **Copy the `main` branch only**

Contribute back to user-workshop-cicd/r.package.example by adding your own branch. Learn more.

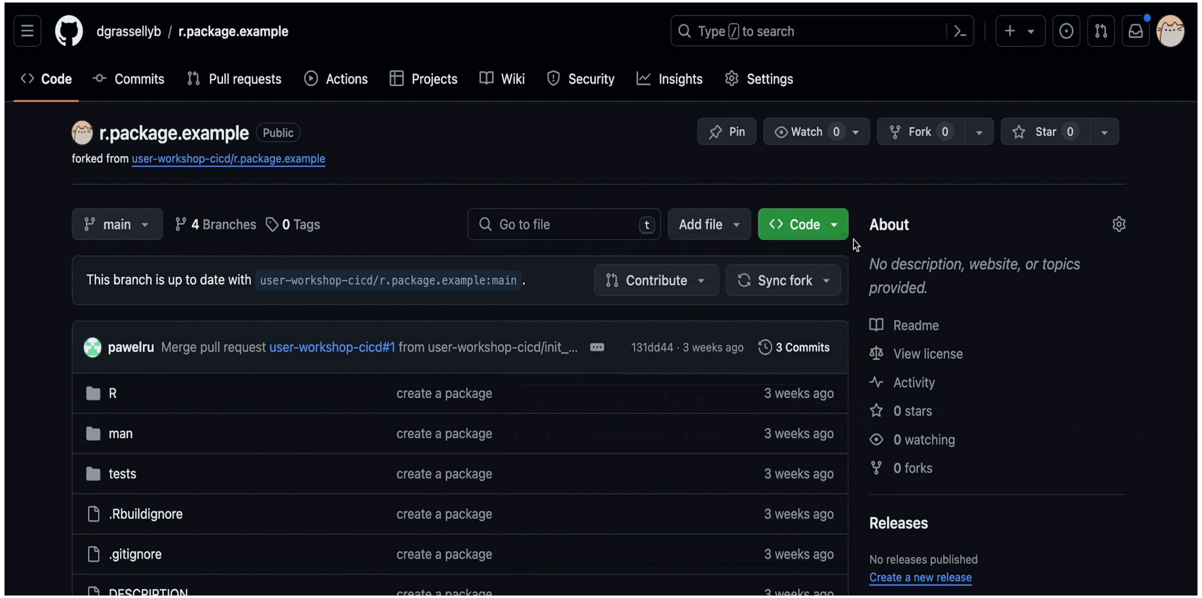
You are creating a fork in your personal account.

Exercises Setup

Cloning with SSH

(Optionally, for advanced users) Clone the forked repositories if you'd like to push commits locally.

Copy SSH URL →



dgrassellyb / r.package.example

Code · Commits · Pull requests · Actions · Projects · Wiki · Security · Insights · Settings

r.package.example Public

forked from user-workshop-cicd/r.package.example

main · 4 Branches · 0 Tags · Go to file · Add file · Code

This branch is up to date with `user-workshop-cicd/r.package.example:main`.

Contribute · Sync fork

pawelru Merge pull request user-workshop-cicd#1 from user-workshop-cicd/init_... 131dd44 · 3 weeks ago · 3 Commits

R	create a package	3 weeks ago
man	create a package	3 weeks ago
tests	create a package	3 weeks ago
.Rbuildignore	create a package	3 weeks ago
.gitignore	create a package	3 weeks ago

About

No description, website, or topics provided.

Readme · View license · Activity · 0 stars · 0 watching · 0 forks

Releases

No releases published

Create a new release

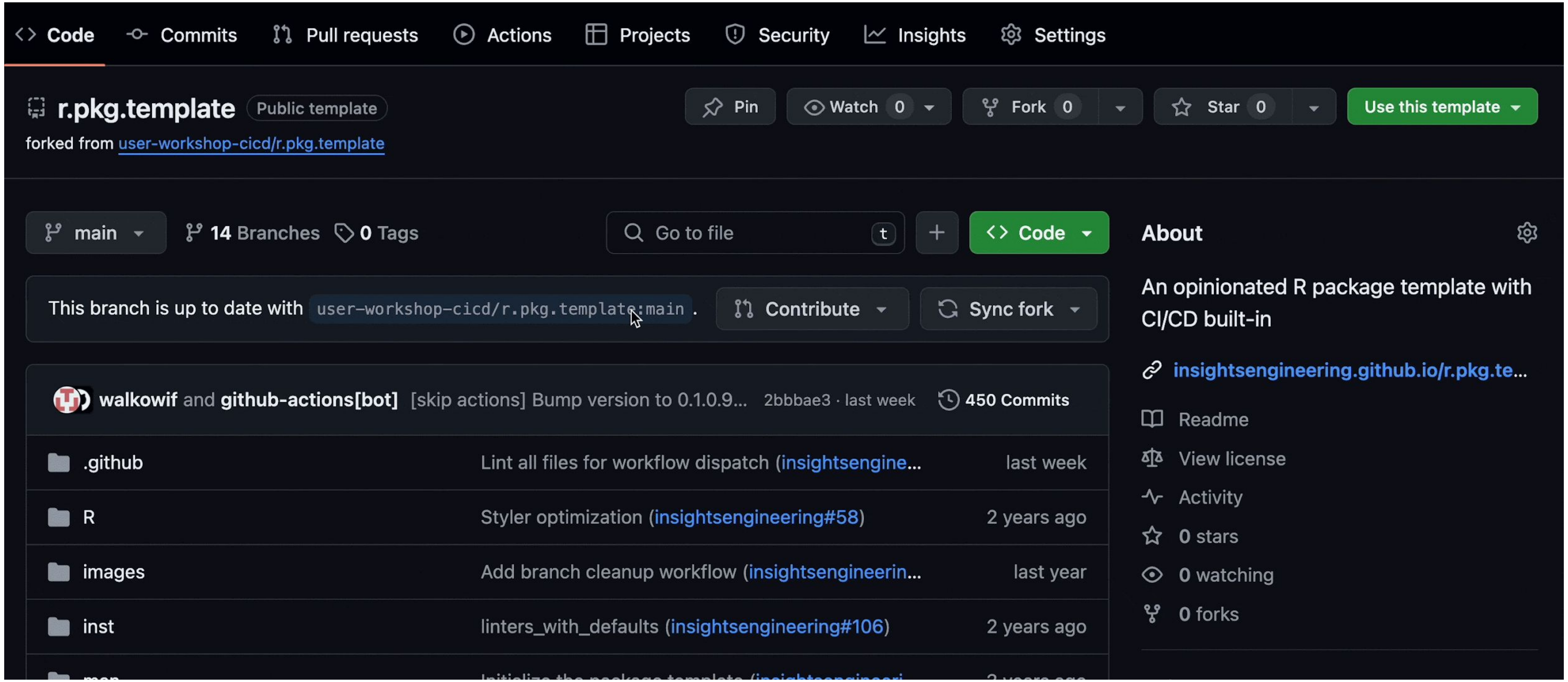
Git clone command →

exercices_use_r — -bash — 125×30

```
grasseld-7M229D:exercices_use_r grasseld$ g
```

Exercises Setup

Enable GitHub workflows: Actions ⟶ I understand my workflows, go ahead and enable them.



forked from user-workshop-cicd/r.pkg.template

This branch is up to date with `user-workshop-cicd/r.pkg.template:main`.

An opinionated R package template with CI/CD built-in

insightsengineering.github.io/r.pkg.te...

walkowif and github-actions[bot] [skip actions] Bump version to 0.1.0.9... 2bbbae3 · last week 450 Commits

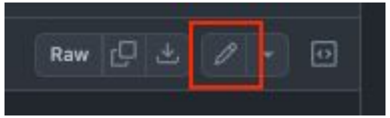
.github	Lint all files for workflow dispatch (insightsengine...	last week
R	Styler optimization (insightsengineering#58)	2 years ago
images	Add branch cleanup workflow (insightsengineerin...	last year
inst	linters_with_defaults (insightsengineering#106)	2 years ago

Exercise 1: Run your first workflow

Exercise 1

Run your first Github workflow!

1. Open forked repo.
2. Let's take a look what is in the repository.
3. Make a dummy commit and push changes.
   a. Open the README.md file.
   b. Click on the pencil button (top right) to introduce changes.
   c. Add an empty line anywhere in the file.
   d. Click on "Commit changes…".
   e. Click on "Commit changes".
4. See an analyse the result in the Actions tab.



Exercise 1

Run your first Github workflow!

1. Go to `github/workflows/simple.yaml` file and find the job called `hello-world-from-r`.
2. Replace “Your name” with your actual name.
   a. Pencil button
   b. Modify
   c. Commit
3. See an analyse the result in the Actions tab.

Exercise 2: Check R package

Introduction

- In this exercise we will be modifying `build-and-check` job.
- This job consists of multiple steps, some of them are having a dummy `if: false` condition that we will be enabling on a step-by-step basis.
- Currently the job is failing and your task is to make this succeed.

Exercise 2a

Set up R environment

~~First of all, we need to install R in order to use it.~~ Actually, the `ubuntu-latest` image already has R pre-installed! Nevertheless, let's make a clean installation on the top of it.

1. Enable "Setup R" step.
2. Analyse the outcome of the following steps.

Exercise 2b

Install package dependencies

R CMD CHECK is failing because of missing dependent packages. We need to install them.

1. Enable “Setup R dependencies” step.
2. Analyse the outcome of the following steps.
   This would make R CMD BUILD and R CMD CHECK executable. Please analyse its results.

Exercise 2c

Fix test error

R CMD CHECK is failing due to test error. Let’s fix it.

1. Navigate to `tests/testthat/test-hello.R` and fix the test error.

Exercise 2d
Upload artifact

It’s pretty common that you want the job to return you some files for further use - debugging or even a separate workflow.

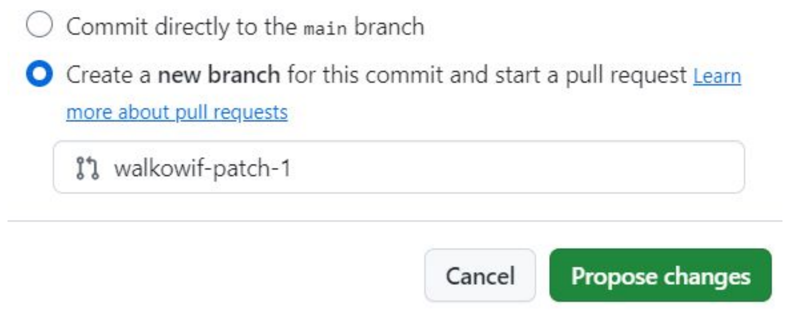
1. Enable “Upload package build” step.

BREAK (15 min)

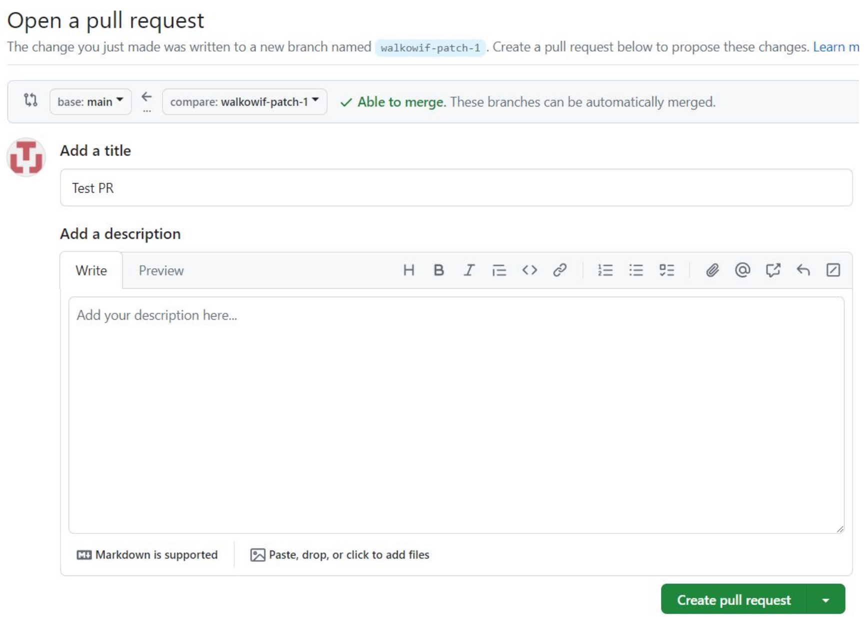
Exercise 3: Triggers

Exercise 3
Triggers

- Edit README.md and Commit changes, but this time…
- Select "Create a **new branch**…" - you can leave the default branch name.
- Click "Propose changes".
- Add a title to the pull request.
- Click "Create pull request".
- Take a look at the checks!


Commit directly to the `main` branch
[✓ Create a **new branch** for this commit and start a pull request] Learn more about pull requests

walkowif-patch-1


Open a pull request

The change you just made was written to a new branch named `walkowif-patch-1`. Create a pull request below to propose these changes. Learn m

base: main ← compare: walkowif-patch-1 ✓ **Able to merge.** These branches can be automatically merged.

Add a title

Test PR

Add a description

Add your description here...

Exercise 3

Triggers

There are many possibilities to trigger a workflow:

- Push new code in a specific branch
- Pushing to a branch from which a pull request has been opened
- Release creation
- Issues creation
- Workflow dispatch: workflows triggered manually
- Scheduled workflows: (cron syntax to configure when your scheduled workflow should run)
- Check the documentation for more details

```
on: workflow_dispatch
```

```
on:
  schedule:
    - cron: '30 5 * * 1,3'
    - cron: '30 5 * * 2,4'
```

Exercise 4: Reusable workflows & job interdependencies

Exercise 4

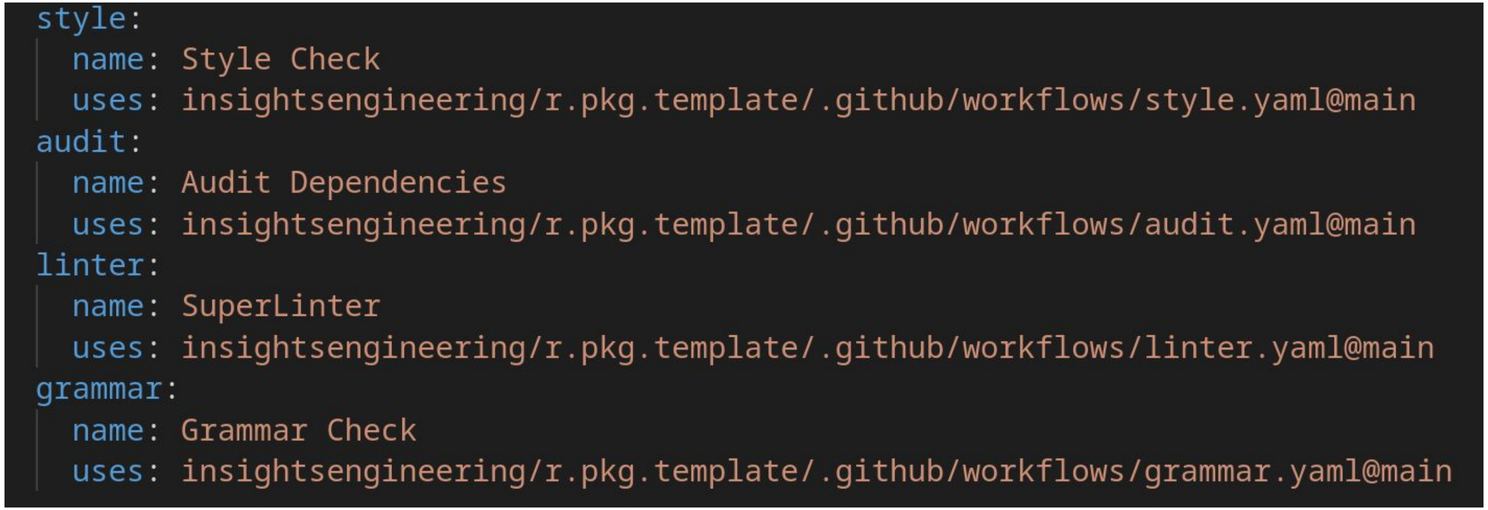
Reusable workflows & job interdependencies

- Open `.github/workflows/check.yaml` file.
- Change "`run_r_cmd_check=false`" to "`run_r_cmd_check=true`"
- What will happen? Why it can be useful? (Complex logic for jobs interdependencies.)
- Explore `.github/workflows/check.yaml` file and find step with reusable workflow.
- Open `.github/workflows/simple.yaml` file.
- Change "Octocat" to your name.
- Find `sample-reusable-workflow.yaml` in https://github.com/user-workshop-cicd/r.pkg.template and explain why it is running now.
- Why reusable workflows? ("Golden standard" of processes to ensure high quality, consistency between various repositories, clarity of custom inputs/parameters used for some repositories).

Exercise 4

Reusable workflows & job interdependencies

- Visit `insightsengineering/r.pkg.template` repository and check workflows which are there.
- Add a new reusable workflow from there to `check.yaml`.



```yaml
style:
  name: Style Check
  uses: insightsengineering/r.pkg.template/.github/workflows/style.yaml@main
audit:
  name: Audit Dependencies
  uses: insightsengineering/r.pkg.template/.github/workflows/audit.yaml@main
linter:
  name: SuperLinter
  uses: insightsengineering/r.pkg.template/.github/workflows/linter.yaml@main
grammar:
  name: Grammar Check
  uses: insightsengineering/r.pkg.template/.github/workflows/grammar.yaml@main
```

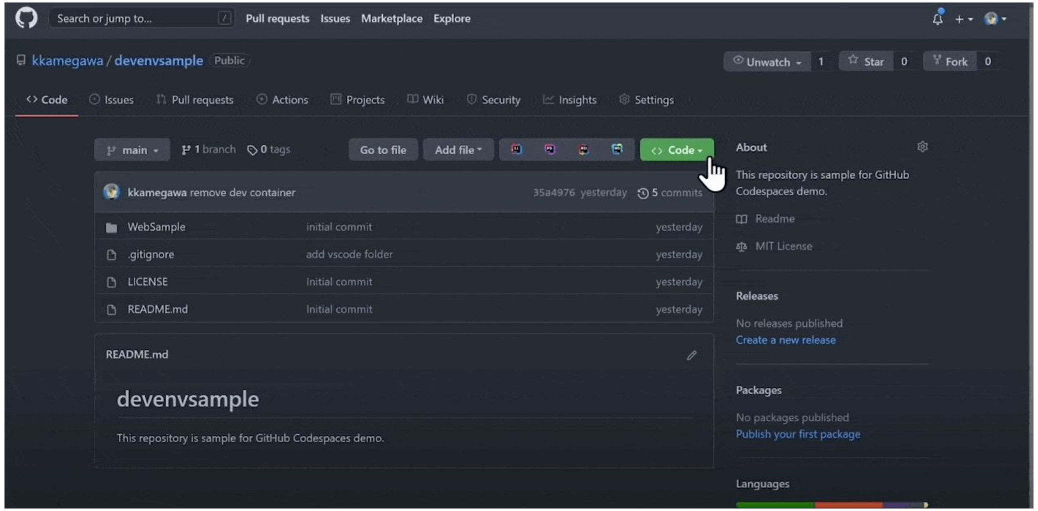
Github Codespaces

Codespaces

Codespaces

What are Codespaces ?

- Development environment **hosted in the cloud** (using github VM infra) - Accessible from your browser
- A codespace can customized and **built on top of a docker image**



kkamegawa remove dev container 35a4976 yesterday 5 commits

WebSample	initial commit	yesterday
.gitignore	add vscode folder	yesterday
LICENSE	Initial commit	yesterday
README.md	Initial commit	yesterday

README.md

devenvsample

This repository is sample for GitHub Codespaces demo.

This repository is sample for GitHub Codespaces demo.

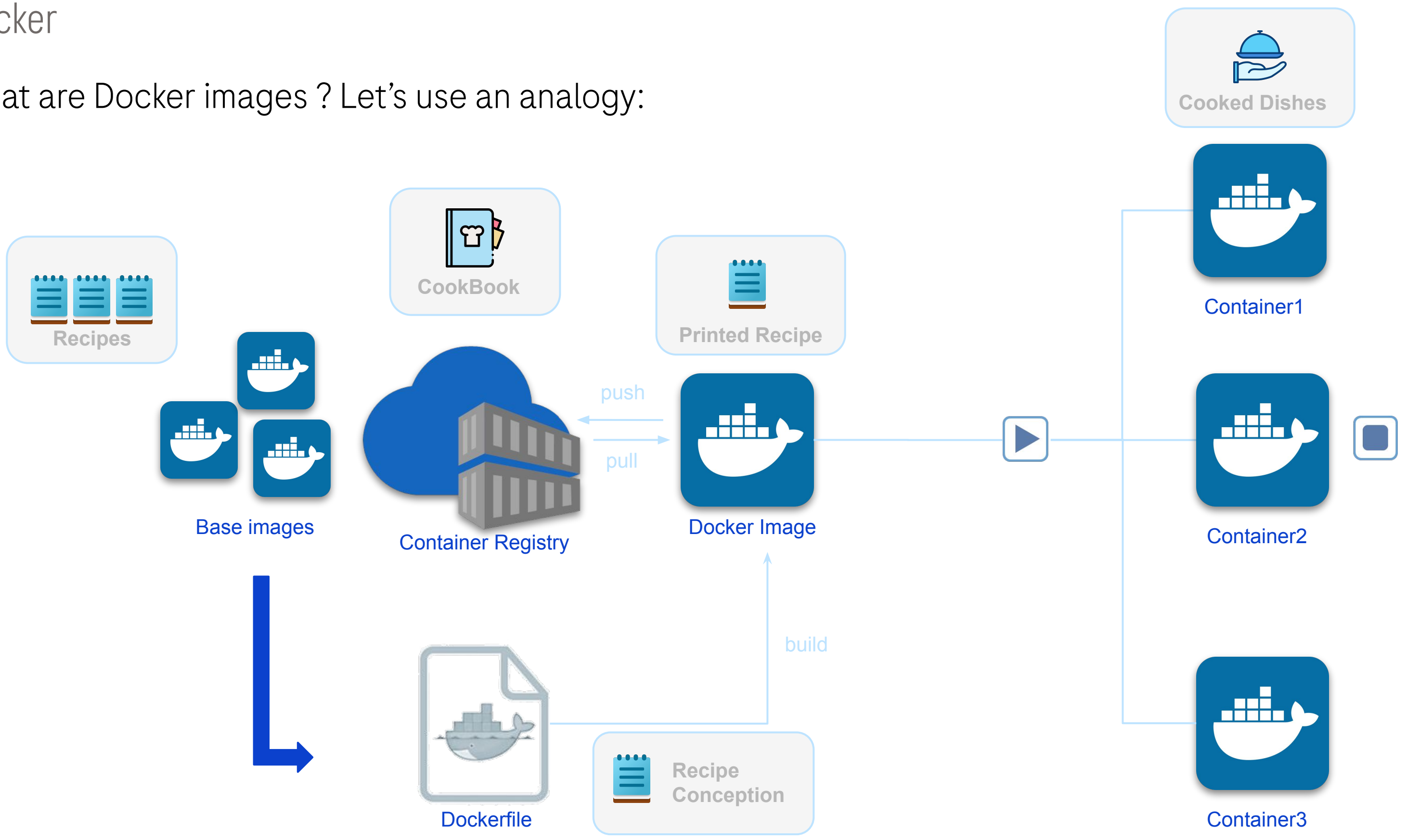
No releases published
Create a new release

No packages published
Publish your first package

Codespaces

Docker

What are Docker images ? Let's use an analogy:



Codespaces

Docker : Rocker Project

Rocker project: set of R images.
Some contains base installation of R,
and some contains other installations,

such as Rstudio Server, tidyverse packages ..

(link to rocker project)

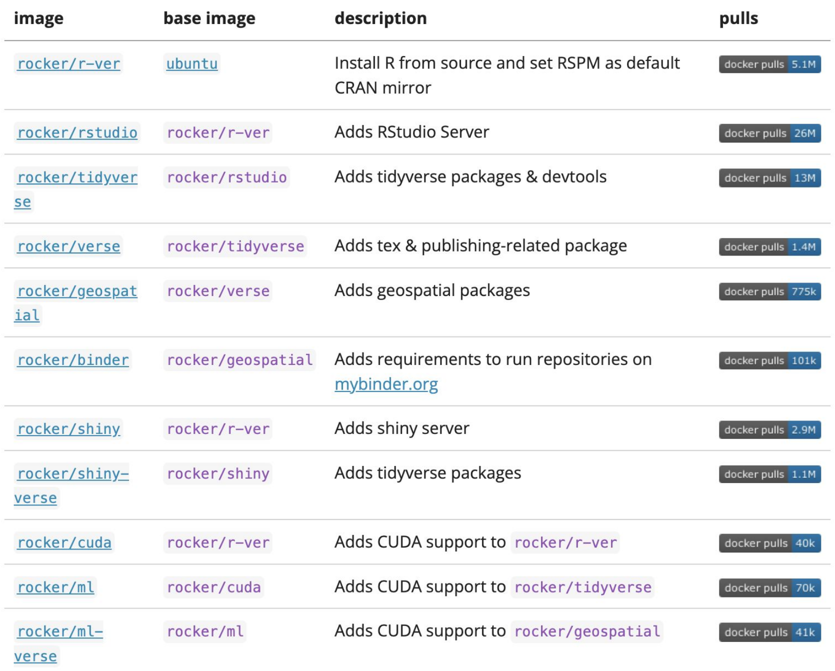
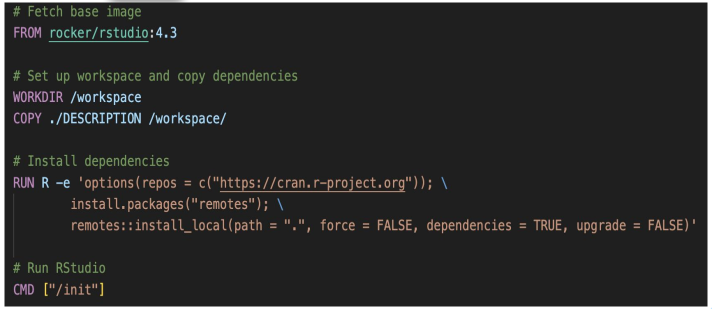


image	base image	description	pulls
rocker/r-ver	ubuntu	Install R from source and set RSPM as default CRAN mirror	docker pulls 5.1M
rocker/rstudio	rocker/r-ver	Adds RStudio Server	docker pulls 26M
rocker/tidyverse	rocker/rstudio	Adds tidyverse packages & devtools	docker pulls 13M
rocker/verse	rocker/tidyverse	Adds tex & publishing-related package	docker pulls 1.4M
rocker/geospatial	rocker/verse	Adds geospatial packages	docker pulls 775k
rocker/binder	rocker/geospatial	Adds requirements to run repositories on mybinder.org	docker pulls 101k
rocker/shiny	rocker/r-ver	Adds shiny server	docker pulls 2.9M
rocker/shiny-verse	rocker/shiny	Adds tidyverse packages	docker pulls 1.1M
rocker/cuda	rocker/r-ver	Adds CUDA support to rocker/r-ver	docker pulls 40k
rocker/ml	rocker/cuda	Adds CUDA support to rocker/tidyverse	docker pulls 70k
rocker/ml-verse	rocker/ml	Adds CUDA support to rocker/geospatial	docker pulls 41k

Codespaces

Configure codespaces using custom docker image

Dockerfile



```
# Fetch base image
FROM rocker/rstudio:4.3

# Set up workspace and copy dependencies
WORKDIR /workspace
COPY ./DESCRIPTION /workspace/

# Install dependencies
RUN R -e 'options(repos = c("https://cran.r-project.org")); \
        install.packages("remotes"); \
        remotes::install_local(path = ".", force = FALSE, dependencies = TRUE, upgrade = FALSE)'

# Run RStudio
CMD ["/init"]
```

Github Workflow

GHCR (Docker Container Registry)

GitHub Codespace Configuration

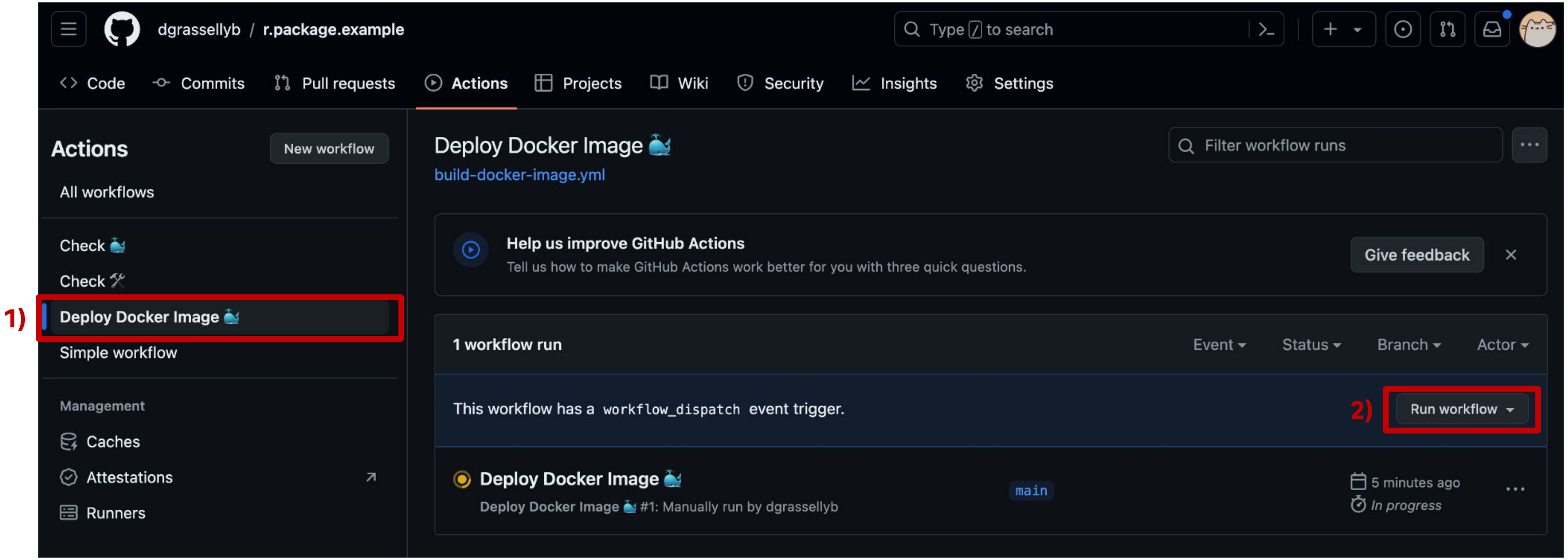
Json configuration

Built on top of **r.package.example** image

- Note : It's possible to configure several codespaces configurations (for example several configurations for several R versions)

Run deploy-image workflow

(Prerequisite for next exercices - Deploy docker image in your user package registry)



Deploy Docker Image 🐳

build-docker-image.yml

Help us improve GitHub Actions

Tell us how to make GitHub Actions work better for you with three quick questions.

1 workflow run

This workflow has a `workflow_dispatch` event trigger.

Deploy Docker Image 🐳

Deploy Docker Image 🐳 #1: Manually run by dgrassellyb

5 minutes ago

In progress

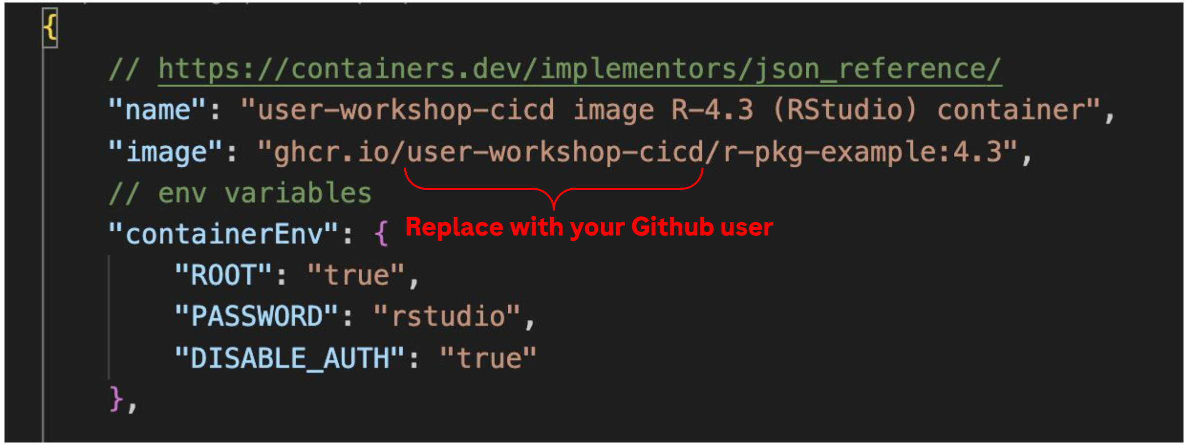
BREAK (15 min)

Exercise 5 (Bonus) : Play with Codespaces

Exercice 5

Now let's play with codespaces !

To start playing with codespaces in your forked repository, you'll need to update the codespace configuration (file **.devcontainers/devcontainer.json**) :



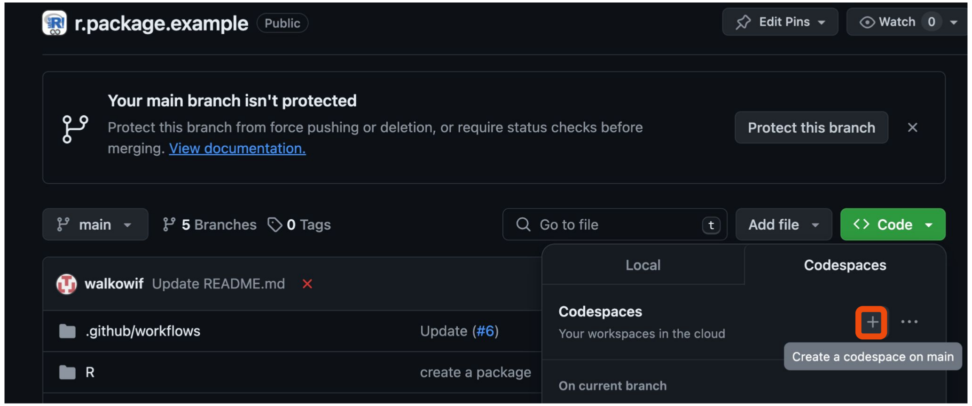
```
{
    // https://containers.dev/implementors/json_reference/
    "name": "user-workshop-cicd image R-4.3 (RStudio) container",
    "image": "ghcr.io/user-workshop-cicd/r-pkg-example:4.3",
    // env variables
    "containerEnv": {
        "ROOT": "true",
        "PASSWORD": "rstudio",
        "DISABLE_AUTH": "true"
    },
```

There is a docker image already available in your forked repository, you can't use image located in user-workshop-cicd for security reasons.

Exercice 5

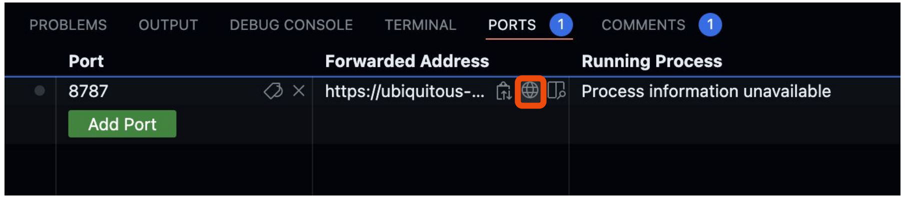
Now let's play with codespaces !

Click here to launch a new codespace from your forked repository :



r.package.example
Public
Edit Pins
Watch 0
Your main branch isn't protected
Protect this branch from force pushing or deletion, or require status checks before merging. View documentation.
Protect this branch
main
5 Branches
0 Tags
Go to file
Add file
Code
Local
Codespaces
walkowif Update README.md
.github/workflows Update (#6)
R create a package
Codespaces
Your workspaces in the cloud
Create a codespace on main
On current branch

Then Open Rstudio IDE (Pay attention to **deactivate ads blockers** !) :



Port	Forwarded Address	Running Process
8787	https://ubiquitous-...	Process information unavailable

Exercice 5

Now let’s play with codespaces !

Play inside the terminal (running R CMD checks, visualizing pre-installed deps ..)

- *setwd("/workspaces/r.package.example")*
- *Then change the workdir of Rstudio*
- *Run installed.packages()*
- *R CMD build --no-manual --no-build-vignettes .*

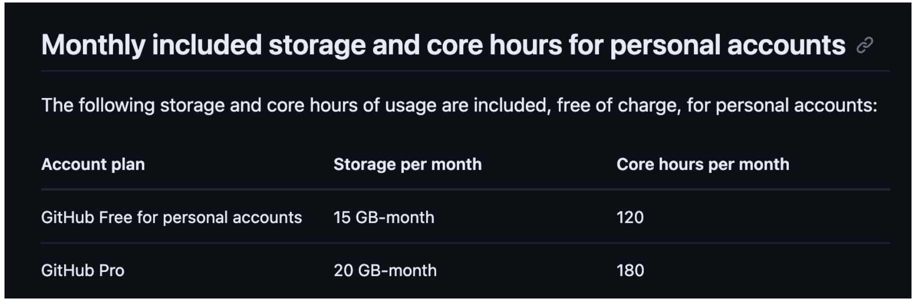
Pay attention to use options *--no-manual --no-build-vignettes* for R CMD build/check

(To build vignettes we need Latex dependency which is not installed in our docker image)

Codespaces

Monitor your codespaces

- To visualize your codespaces, you can navigate under https://github.com/codespaces
- You can pause a codespace, delete them ..
- Pay attention to the usage limit :



Monthly included storage and core hours for personal accounts

The following storage and core hours of usage are included, free of charge, for personal accounts:

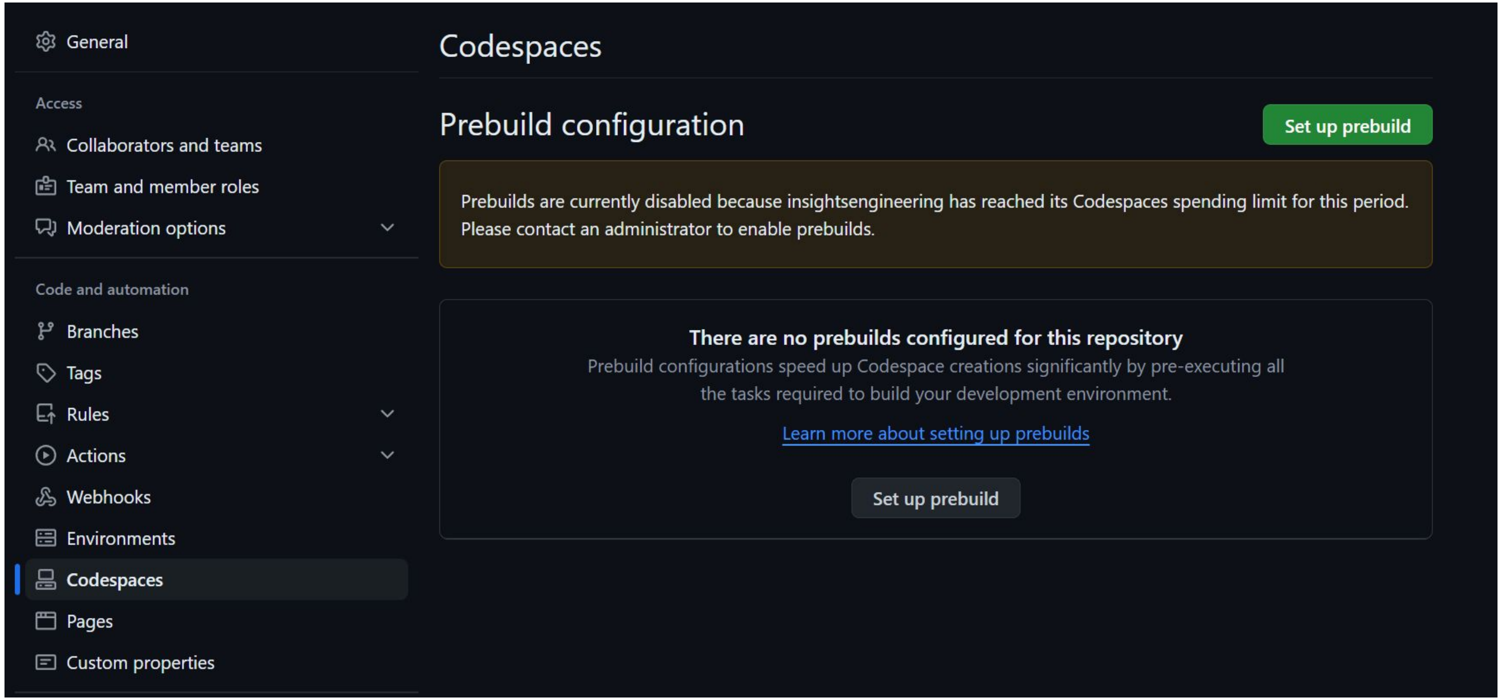
Account plan	Storage per month	Core hours per month
GitHub Free for personal accounts	15 GB-month	120
GitHub Pro	20 GB-month	180

You can see your usage here : https://github.com/settings/billing/summary

Codespaces - Pre Build settings

Pre-build a codespace (at the end of this tutorial)

To save time when building the codespaces for a repository end users, it's possible to pre-build codespaces. Let's do this ! Go under **Settings -> Codespaces** and start your pre-build config :



Prebuilds are currently disabled because insightsengineering has reached its Codespaces spending limit for this period. Please contact an administrator to enable prebuilds.

There are no prebuilds configured for this repository

Prebuild configurations speed up Codespace creations significantly by pre-executing all the tasks required to build your development environment.

Learn more about setting up prebuilds

Set up prebuild

Pre-build time :

~ 5min

Exercise 6 (Bonus) : Replace Github runner by container

Exercice 6 (Bonus)

Replace Github runner by container

Enable job **r-cmd-with-docker** (file **check-using-docker.yaml**)

1. Update user-workshop-ci by your Github user in image ghcr.io/user-workshop-cicd/r-pkg-example:4.3 (Like we did for codespaces config file)
2. Look at R CMD checks logs

Question (a bit difficult this one !) :

- With jobs running on container, it’s not possible to use reusable workflows. Why ?

Roche

Questions and Discussions

Doing now what patients need next