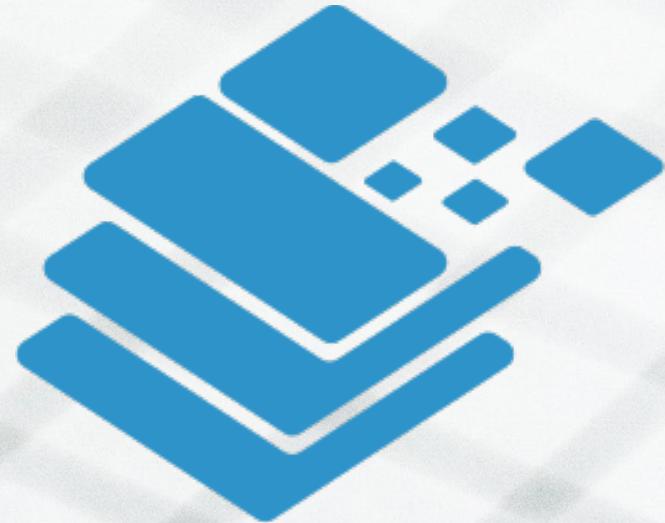


partimos pronto



**Spatial
Lab
Analytics**

Soluciones en análisis de datos

APLICACIONES INTERACTIVAS CON R Y SHINY

Bastián Olea Herrera - baolea@uc.cl

Desarrollo de interfaces web para interactuar con código de R en tiempo real, permitiendo que analistas y usuarios accedan a procesos, resultados y visualizaciones de R mediante su navegador.

01

Aplicación local

Compartir una aplicación que se ejecute localmente en el equipo de otra persona.

02

ShinyApps.io

Publicar la aplicación en el servicio en la nube de Posit

03

DigitalOcean

Montar la aplicación en un servidor privado escalable



The screenshot shows the RStudio interface with the 'New Project Wizard' dialog open. The dialog has three main options: 'New Directory', 'Existing Directory', and 'Version Control'. Each option includes a brief description and a right-pointing arrow indicating further steps. The 'New Directory' option is currently selected. The background of the RStudio window shows the environment pane with tabs for Environment, History, Connections, and Tutorial, and a global environment table.

New Project Wizard

Create Project

New Directory
Start a project in a brand new working directory >

Existing Directory
Associate a project with an existing working directory >

Version Control
Checkout a project from a version control repository >

Cancel

01

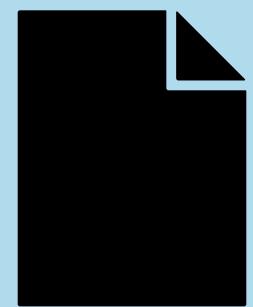
app local

Compartir una aplicación que se ejecute localmente en el equipo de otra persona.

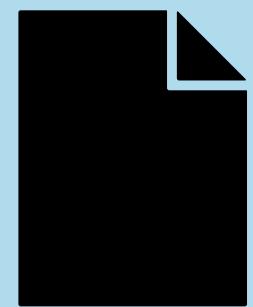


Confirmar que la app funciona localmente

- Confirma que estás trabajando con Proyectos de RStudio
- Confirma que el script de tu app está dentro del proyecto
- Reinicia tu sesión de R para confirmar que tu app es reproducible



app.R



funciones.R



datos



Crear un entorno reproducible

- Para asegurar que tu aplicación funcione hacia el futuro, debemos asegurarnos de registrar los paquetes que usemos en el proyecto
- El paquete `renv` crea un **entorno de R reproducible para tu proyecto**, registrando las versiones de los paquetes que usaste



Paquetes

- Al instalar paquetes desde la consola, un script o un proyecto, se instalan en la biblioteca (*library*) de tu computador.
- En la biblioteca de tu computador están todos los paquetes
- Cuando cargas un paquete desde cualquier proyecto, se cargan desde una biblioteca común (la biblioteca de tu computador)
- Por eso, si instalas un paquete en un script, en otro script no necesitas reinstalarlo
- Si actualizas un paquete en un proyecto, otros proyectos que usen el paquete luego cargarán la versión actualizada

```
install.packages("dplyr")
```

instalar

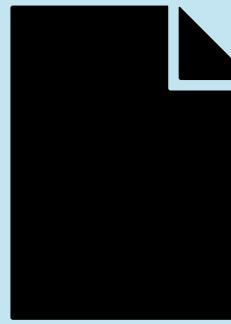
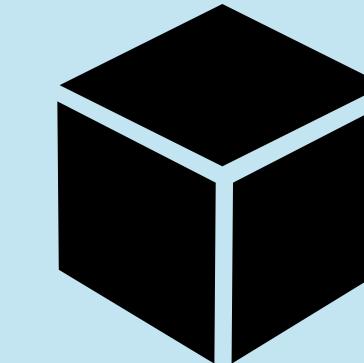
biblioteca

dplyr 1.1.4
stringr 1.5.1
tidy 1.3.1

arrow 17.0.0.1
shiny 1.9.1
ggplot2 3.5.1

cargar

```
library(dplyr)
```



Gestión de entornos con {renv}

- {renv} registra los paquetes que usas
- Permite que los paquetes que instalas o actualices en un proyecto no interfieran con los de tus demás proyectos
- Al instalar la aplicación en otro equipo, {renv} facilita instalar los paquetes necesarios
- {renv} instalará exactamente las versiones de los paquetes que registraste, evitando conflictos de versiones

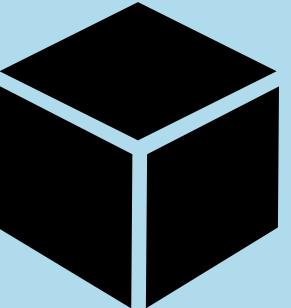
```
install.packages("renv")
library(renv)

# usar {renv} en el proyecto
renv::init()

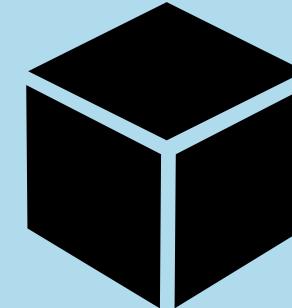
# registrar paquetes usados
renv::snapshot()

# reinstalar paquetes
renv::restore()
```

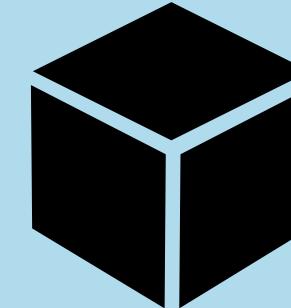
biblioteca 1



biblioteca 2

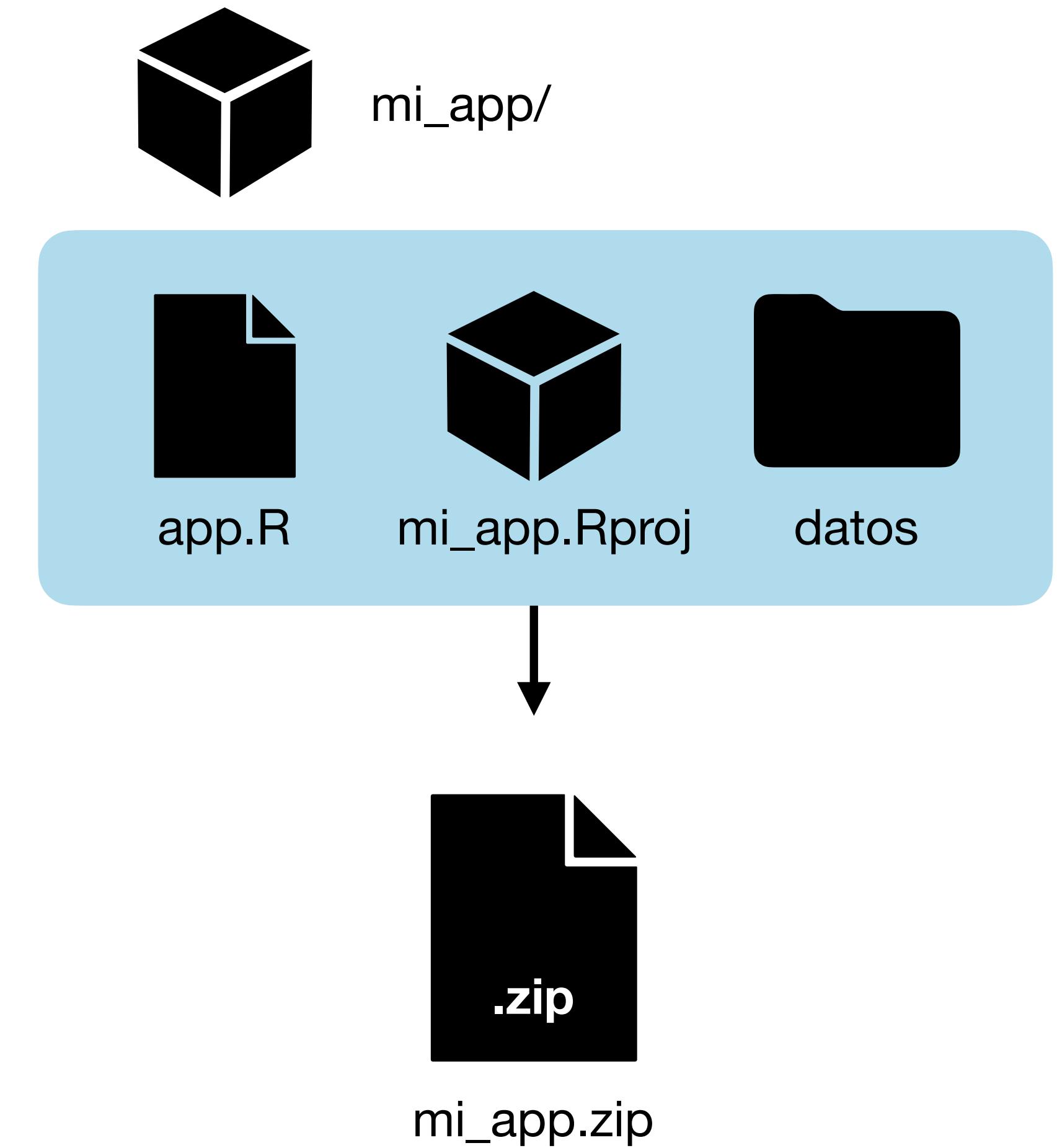


biblioteca 3



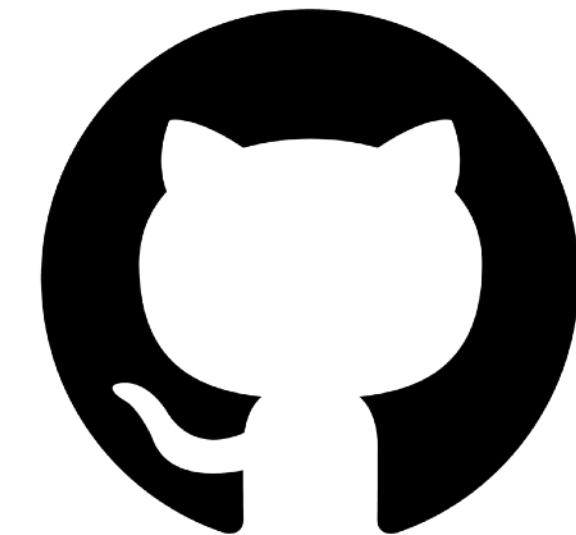
Compartir una app: manualmente

- Si tu aplicación está en un proyecto, puedes **comprimir la carpeta del proyecto** y enviársela a alguien. Esa persona descomprimirá la carpeta del proyecto, y podrá abrir el archivo .Rproj dentro de la carpeta para abrir el proyecto en RStudio.
- El/la otro/a usuario/a deberá **instalar los paquetes correspondientes** en su equipo para poder usar la aplicación.
- Si configuraste {renv}, el usuario/a podrá ejecutar `renv::restore()` para que {renv} lea el *lockfile* e instale los paquetes que corresponden, con las versiones específicas que se usaron.



Compartir una app: GitHub

- Instalar git en tu equipo
<https://git-scm.com/downloads>
- Crear una cuenta en GitHub, si es que no tienes
<https://github.com/>



GitHub

Compartir una app: GitHub

- Primero se debe configurar git en tu equipo. Git es un sistema de control de cambios, que permite ir guardando localmente las versiones de tus proyectos. También permite conectarte a repositorios remotos, para poder almacenar tus versiones online en GitHub, y por consiguiente, poder compartir tus proyectos con otros/as.

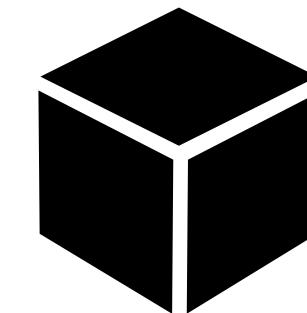
- En Terminal (macOS o Linux):

```
git config --global user.name "bastianolea"  
git config --global user.email baolea@uc.cl
```

- En Windows, lo mismo pero en el archivo:

```
C:\Users\$USER\.gitconfig
```

- Crear una cuenta en GitHub, si es que no tienes



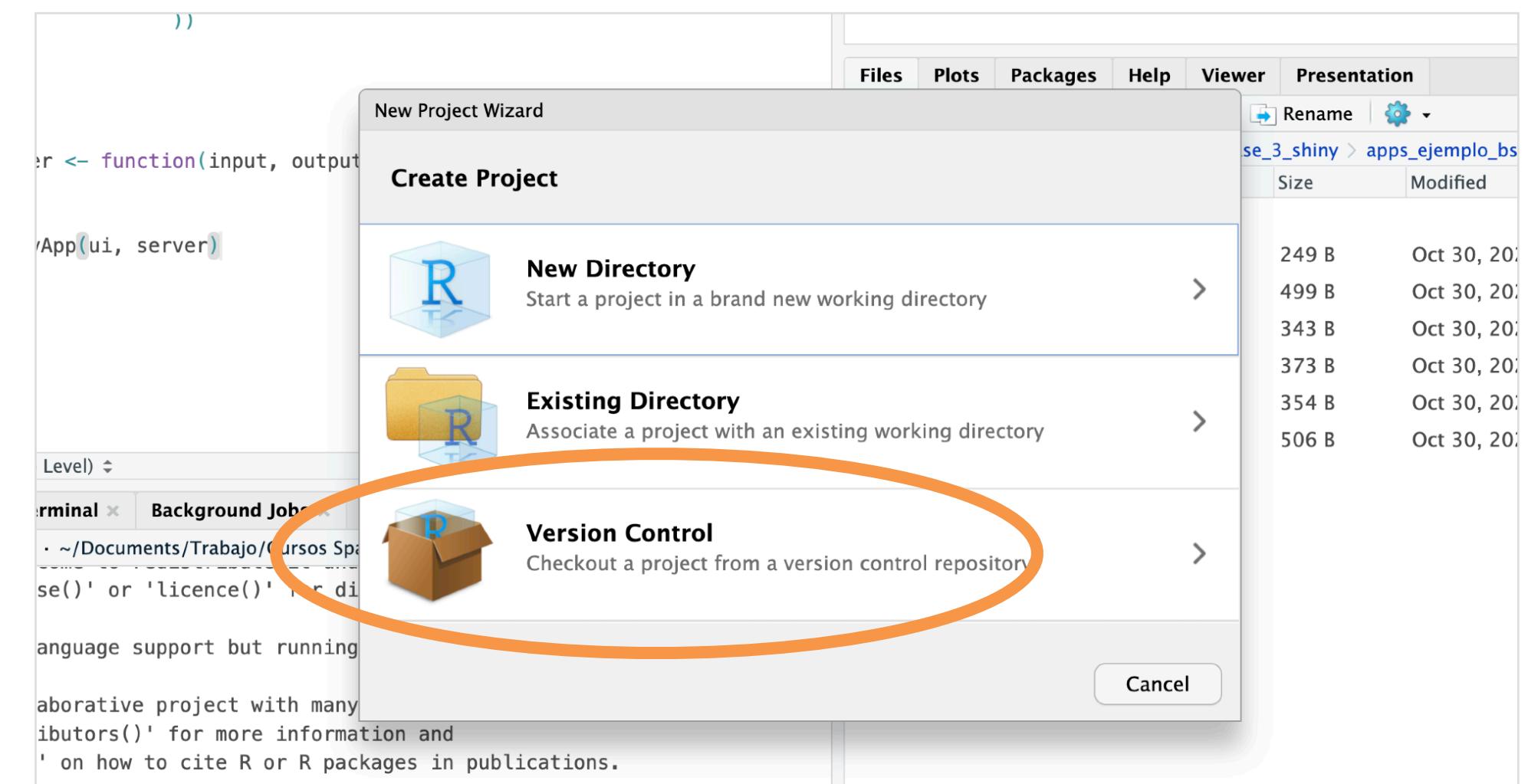
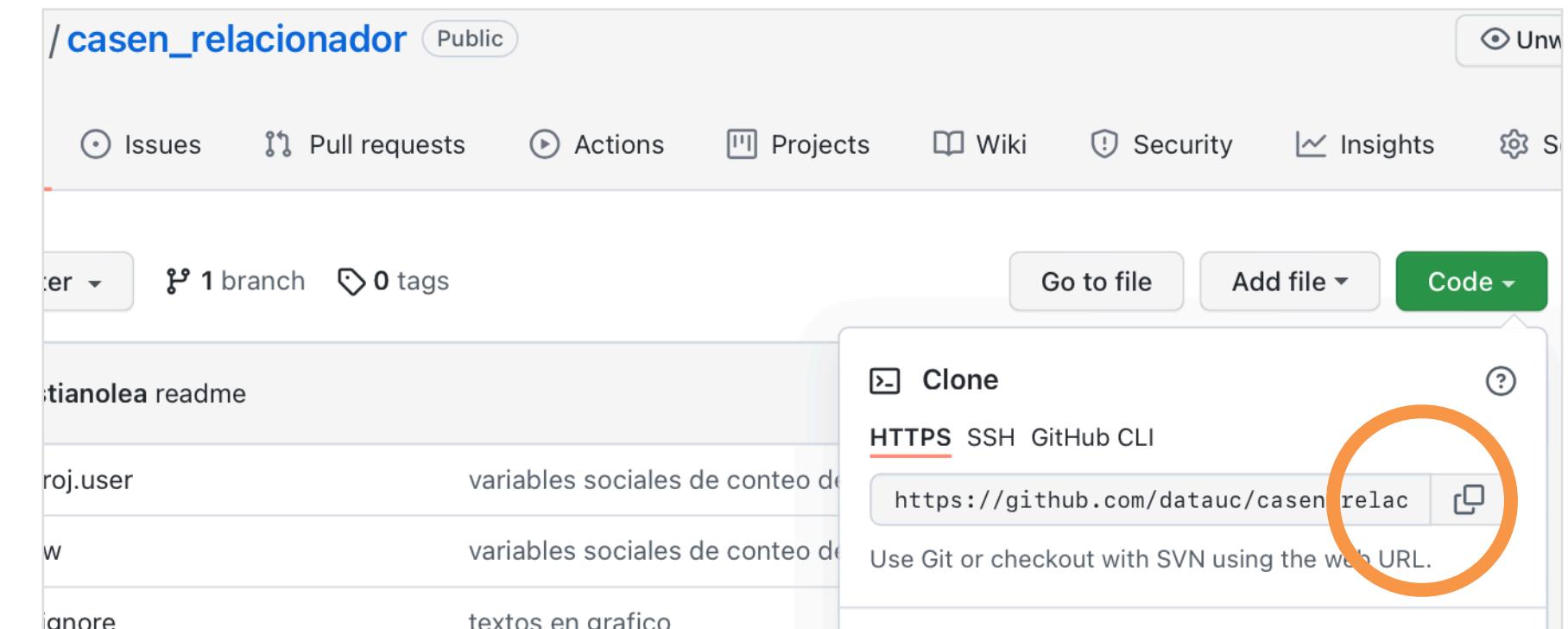
mi_app/

```
# crear repositorio local  
usethis::use_git()  
  
# abre ventana de github, generar y  
copiar token  
usethis::create_github_token()  
  
# ingresar el token  
gitcreds::gitcreds_set()  
  
# crear repositorio remoto y  
conectarlo con el local  
usethis::use_github()
```

https://github.com/bastianolea/economia_chile

Compartir una app: GitHub

- Entrar al repositorio de tu app, o el de la app que deseas clonar en tu equipo
- Copiar su enlace HTTPS
- En RStudio: nuevo proyecto: crear un proyecto desde control de versiones (GitHub)
- Pegar en enlace y definir dónde se ubicará





Id	Name	Status	Instances
12623038	prensa_chile	Sleeping	1
13180950	norma_compensatoria	Sleeping	1
11049199	corrupcion_chile	Sleeping	1
10926798	casen_relacionador	Sleeping	1
11475750	femicidios_chile	Sleeping	1
12900193	iclac_encuesta	Sleeping	1
12292552	economia_chile	Sleeping	1
12373647	estimador_ingresos_trabajo	Sleeping	1
12373649	estimador_ingresos_clases	Sleeping	1
10873336	autonatotattoo	Sleeping	1



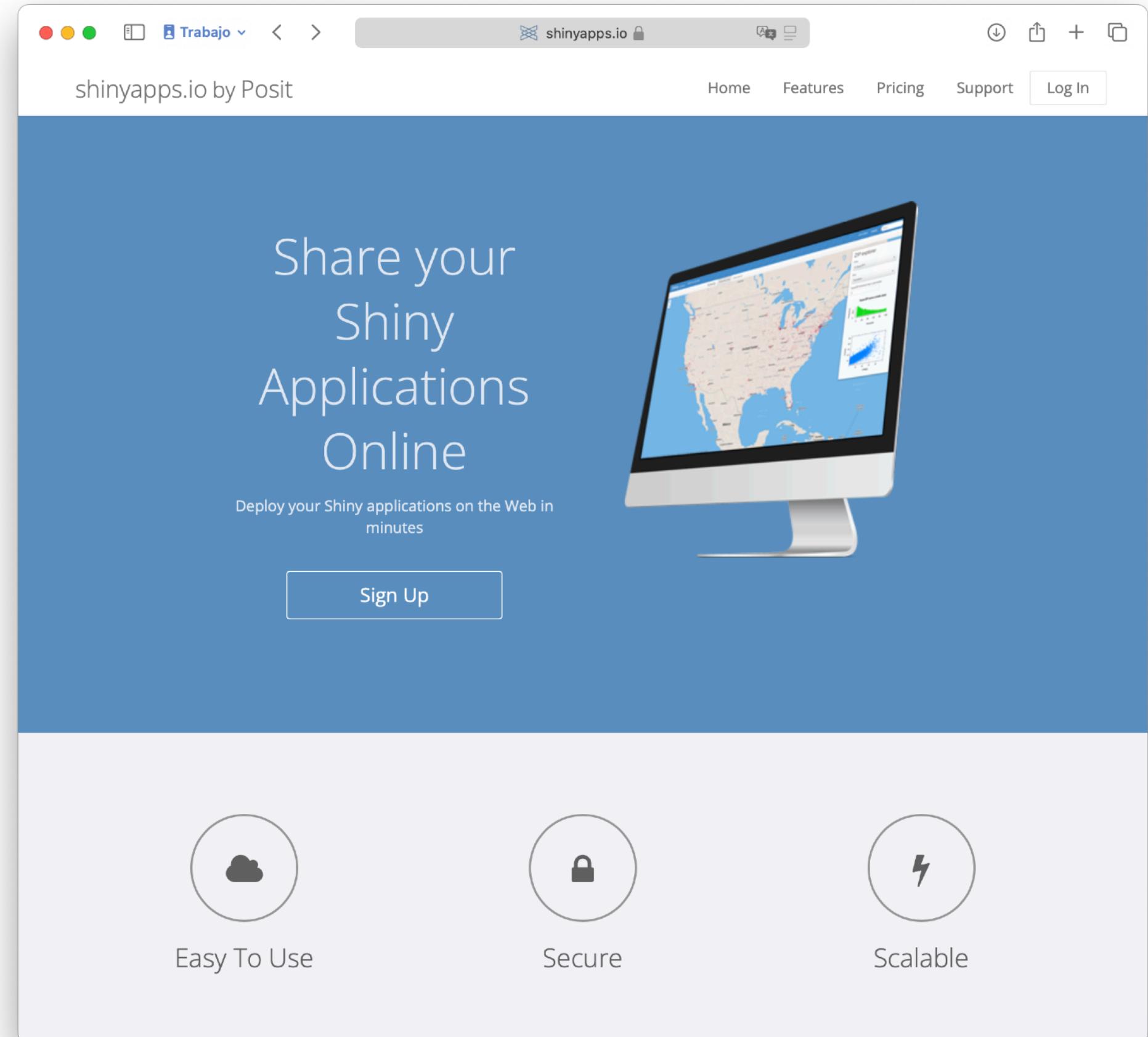
02

ShinyApps.io

Publicar la aplicación en el servicio en la nube de Posit.

Crear una cuenta

- Crear una cuenta en [ShinyApps.io](#)
- Las cuentas gratuitas permiten tener hasta 5 aplicaciones, con 25 horas de uso mensual (en total, entre todos tus usuarios)
- El plan inicial (13USD) permite 25 aplicaciones y 1000 horas de uso mensual.



Conectar cuenta a RStudio

- Instalar el paquete {rsconnect}
- Autorizar tu cuenta de ShinyApps.io en RStudio.
- Para ello, shinyapps te entrega un código secreto asociado a tu cuenta, el cual debes registrar en tu sesión de R

[Start with R](#) [Start with Python](#)

STEP 1 – INSTALL RSCONNECT

The `rsconnect` package can be installed directly from CRAN. To make sure you have the latest version run following code in your R console:

```
install.packages('rsconnect')
```

STEP 2 – AUTHORIZE ACCOUNT

The `rsconnect` package must be authorized to your account using a token and secret. To do this, click the copy button below and we'll copy the whole command you need to your clipboard. Paste it into your R console to authorize your account. Once you've entered the command successfully in R, that computer is now authorized to deploy Shiny for R applications to your shinyapps.io account.

```
rsconnect::setAccountInfo(name='baolea',  
token='ED89E37181991AACC5D69BD',  
secret='<SECRET>')
```

[Show Secret](#) [Copy to clipboard](#)

In the future, you can manage your tokens from the [Tokens](#) page the settings menu.

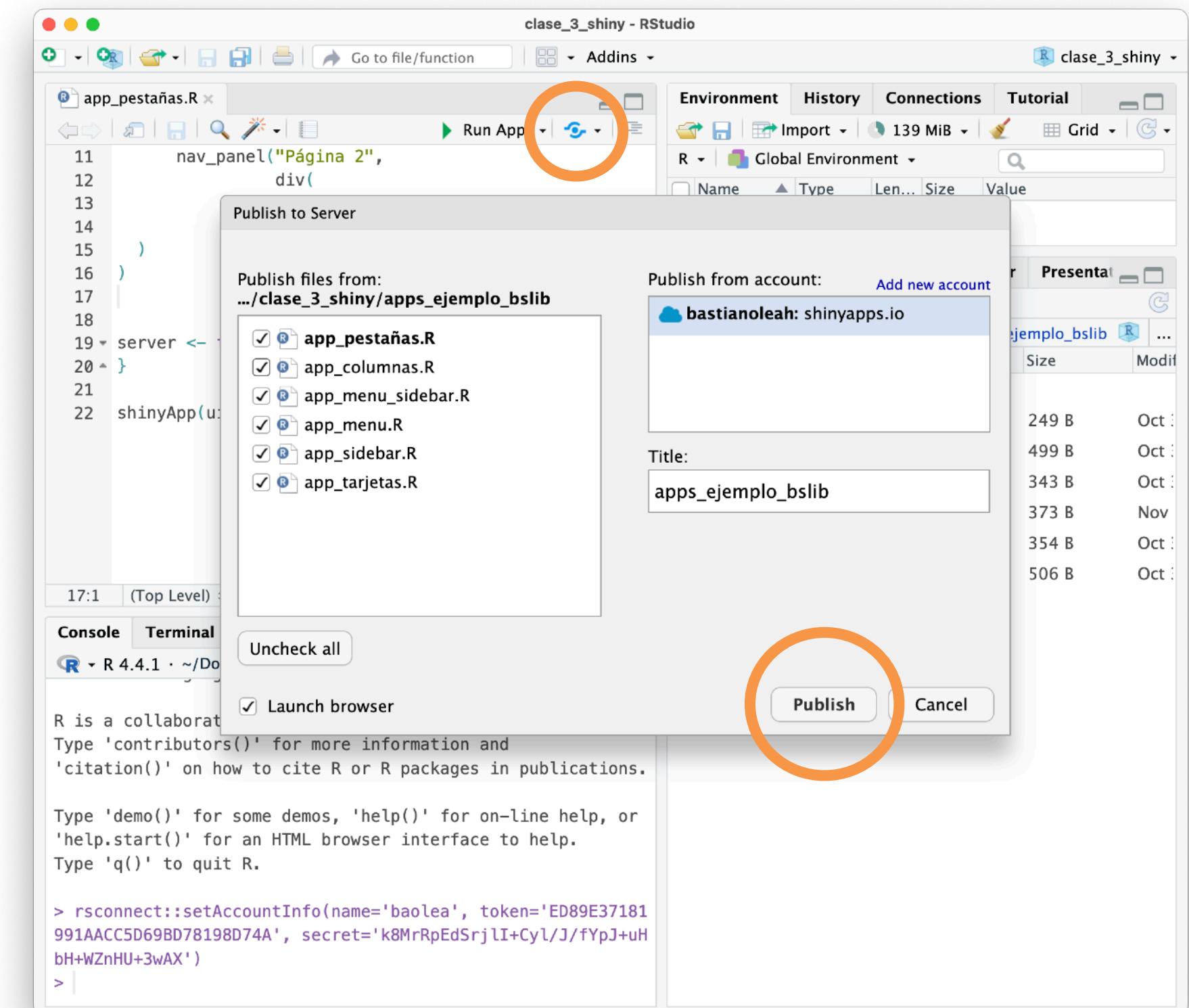
STEP 3 – DEPLOY

Once the `rsconnect` package has been configured, you're ready to deploy your first application. If you haven't written any applications yet, you can also checkout the [Getting Started Guide](#) for instructions on how to deploy our demo application. Run the following code in your R console.

```
library(rsconnect)  
rsconnect::deployApp('path/to/your/app')
```

Publicar apps desde RStudio

- En el script de tu aplicación Shiny, presiona el botón azul de *Publicar*
- Selecciona los archivos a incluir, tu cuenta, y el nombre de la aplicación
- Presiona *Publicar*



Acceder a apps

<https://tucuenta.shinyapps.io/nombredelaapp/>



PROJECTS ^

 first-project

+ New Project

MANAGE ▾

By DigitalOcean

Billing

Support

Settings

API New

Marketplace ↗

Product Docs ↗

What's New ↗

first-project DEFAULT
Update your project information under Settings

Resources Activity Settings

DROPLETS (1)

- rstudio2021092382onubuntu2004... 159.223.108.86 R C

Create something new

-  Create a Managed Database
Worry-free database management
-  Start using Sp...
Deliver data w...
-  Spin up a Load Balancer
Distribute traffic between multiple Droplets

Build on what you have

-  Build a Node.js application
Set up a Node.js application for production on Ubuntu
-  Go containeri...
Install and use containerized
-  Build a PHP web application
Install Linux, Apache, MySQL, PHP (LAMP) on DigitalOcean 1-Click App



/ spatiallab.cl

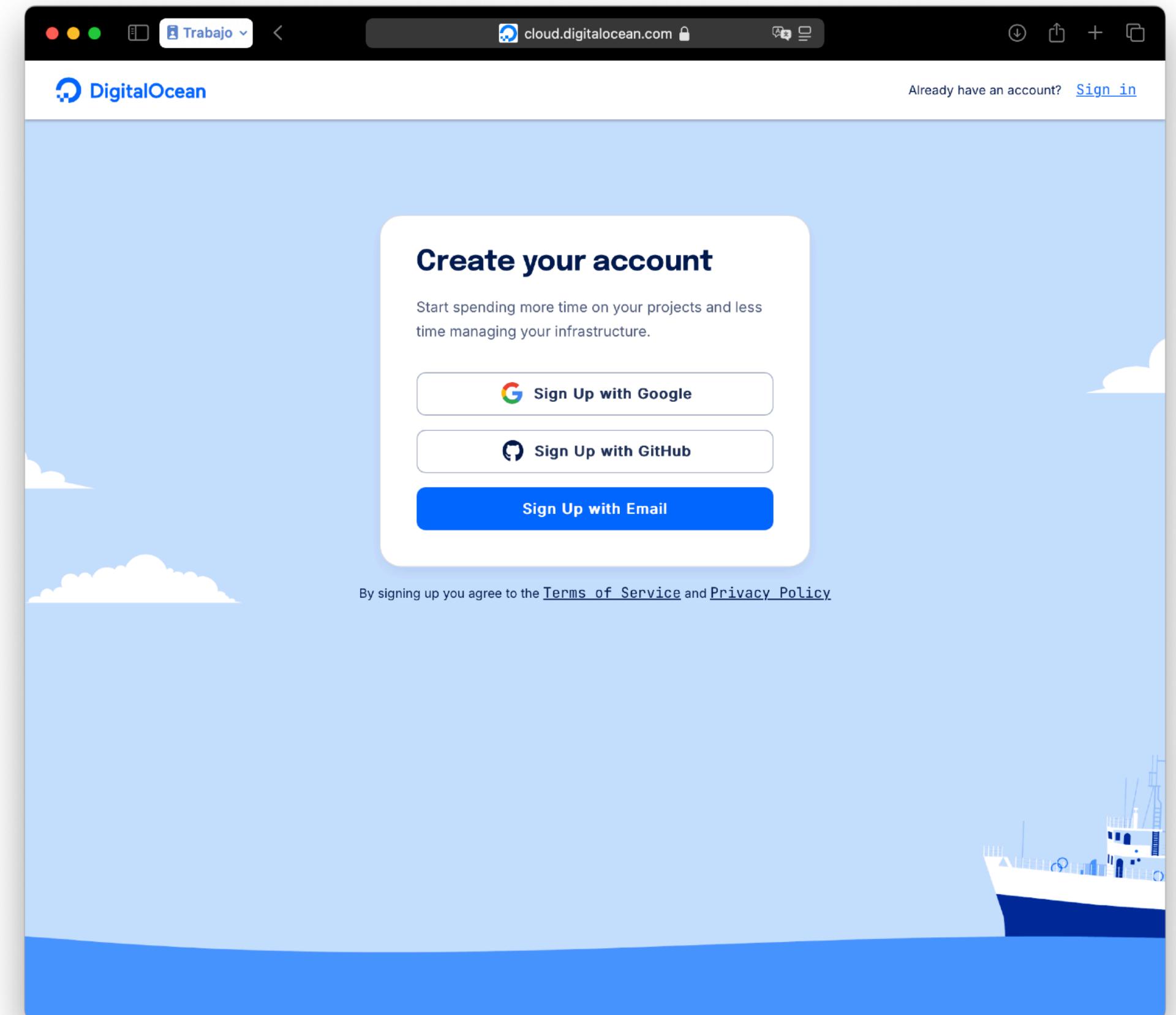
03

Digital Ocean

Montar la aplicación en un servidor privado escalable.

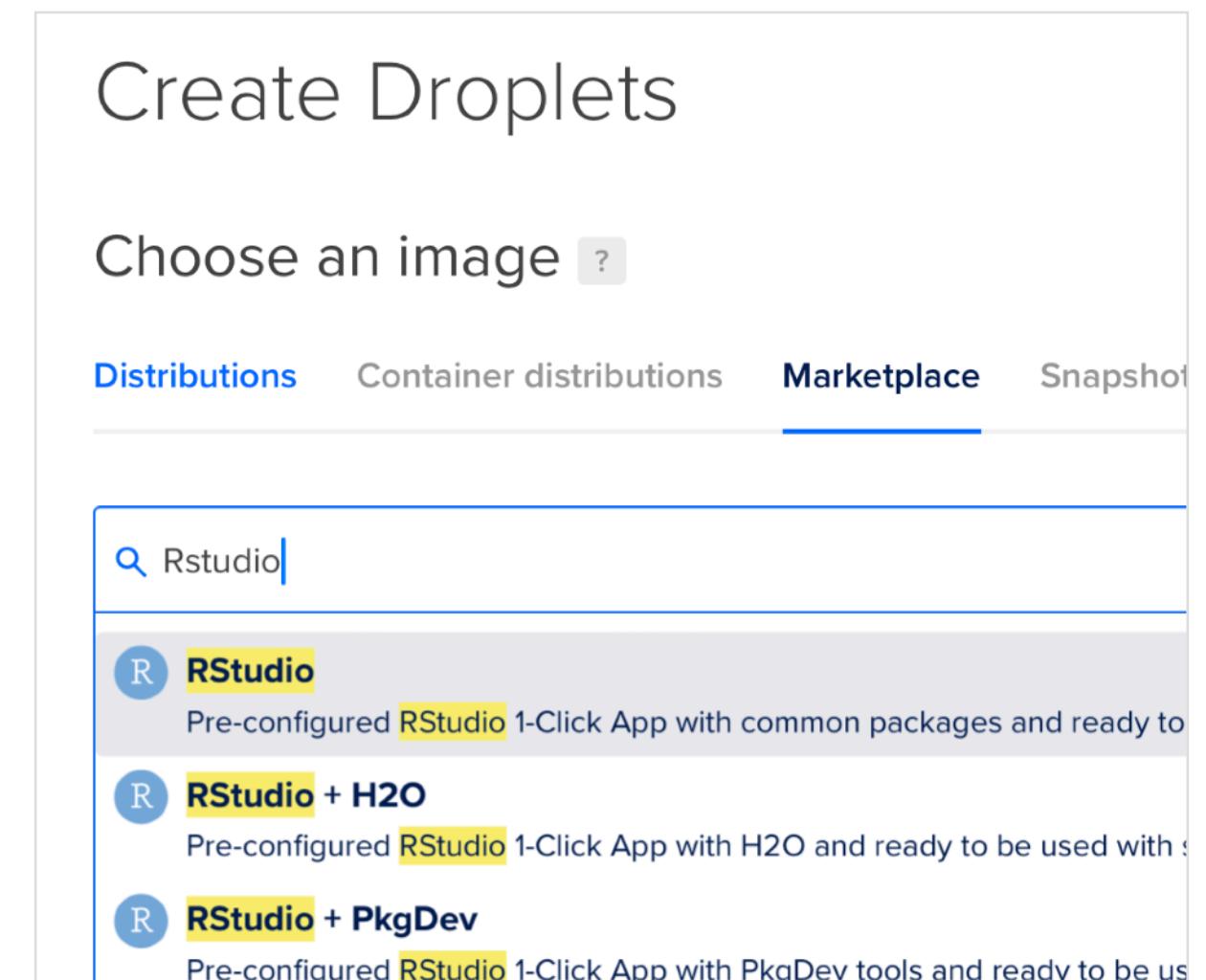
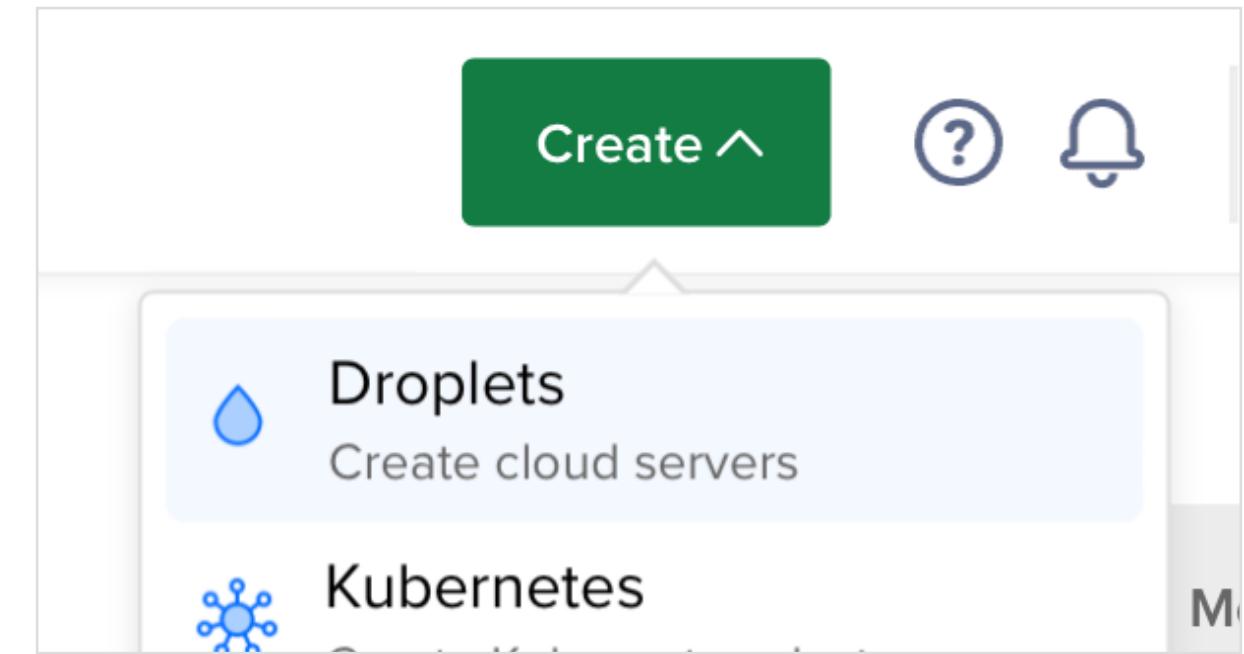
Crear una cuenta

- Crear una cuenta en [DigitalOcean.com](https://cloud.digitalocean.com)
<https://m.do.co/c/b117f791b027>
- Las cuentas son gratuitas, y dependiendo de los recursos que crees, se incurrirá en cobros mensuales
- Puedes crear Droplets (servidores privados virtuales, VPS), Volúmenes (almacenamiento), y más



Crear un droplet

- En el botón superior derecho, crear un Droplet
- Al crear un droplet, se debe elegir la región física donde estará ubicado, el sistema operativo, y el "tamaño" (especificaciones técnicas del servidor)
- Elegimos crear un droplet desde Marketplace, donde existen cientos de imágenes que permiten crear droplets con entornos predefinidos, en este caso, uno con RStudio pre-instalado



Definir configuración

- Más **procesadores** permite mejor velocidad, mejor desempeño de procesos concurrentes (por ejemplo, múltiples apps, tareas recurrentes, etc.)
- Mayor **memoria** permite trabajar con mayores volúmenes de datos y más aplicaciones. R suele caerse si se queda sin memoria.
- Dependiendo de la configuración es el cobro mensual. El cobro mensual considera el uso del recurso 24/7. Si apagamos el recurso, detenemos el cobro.

Choose a plan

SHARED CPU		DEDICATED CPU			
Basic	General Purpose	CPU-Optimized	Memory-Optimized	Storage	
Basic virtual machines with a mix of memory and compute resources. Best for small projects that can handle variable levels of performance, like blogs, web apps and dev/test environments.					
CPU options:	<input checked="" type="radio"/> Regular Intel with SSD	<input type="radio"/> Premium Intel with NVMe SSD <small>NEW</small>	<input type="radio"/> Premium AMD with NVMe SSD <small>NEW</small>		
\$5/mo \$0.007/hour	\$10/mo \$0.015/hour	\$15/mo \$0.022/hour	\$20/mo \$0.030/hour	\$40/mo \$0.060/hour	
1 GB / 1 CPU 25 GB SSD Disk 1000 GB transfer	2 GB / 1 CPU 50 GB SSD Disk 2 TB transfer	2 GB / 2 CPUs 60 GB SSD Disk 3 TB transfer	4 GB / 2 CPUs 80 GB SSD Disk 4 TB transfer	8 GB / 4 CPUs 160 GB SSD Disk 5 TB transfer	



1. Instrucciones

2. IP

3. Consola

4. Apagar

The screenshot shows a digital infrastructure management interface for a droplet named "rstudio2021092382onubuntu2004-s-1vcpu-1gb-nyc1-01".

- 1. Get started:** A blue button labeled "Get started" is highlighted with an orange oval.
- 2. IP:** Shows IPv4: 159.223.108.86, IPv6: Enable now, Private IP: 10.116.0.2, Reserved IP: Enable now. The "Console" link is also highlighted with an orange oval.
- 3. Metrics:** Includes sections for Graphs, Access, Power, Volumes, Resize, Networking, Backups, Snapshots, Kernel, History, Destroy, Tags, and Recovery. A dropdown menu for "Select period" is set to "1 hour".
- 4. Bandwidth:** A chart titled "Bandwidth" showing traffic metrics from 2024-11-06 11:06:30. The chart displays four series:
 - private - inbound: 2.84 b/s (blue line)
 - public - inbound: 334 kb/s (green line)
 - private - outbound: 4.27 b/s (yellow line)
 - public - outbound: 2.30 kb/s (purple line)The chart shows a significant increase in public inbound traffic starting around 11:05.

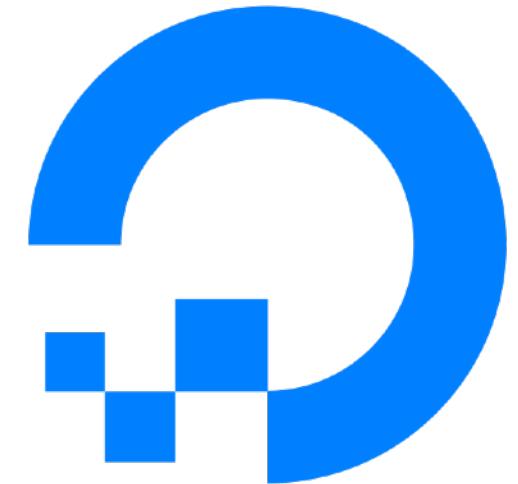


Instrucciones Digital Ocean

- Acceder a la consola del Droplet
- Crear un usuario root:
`adduser rstudio`

- Crear un usuario personal:
`adduser usuario`
`usermod -aG sudo usuario`

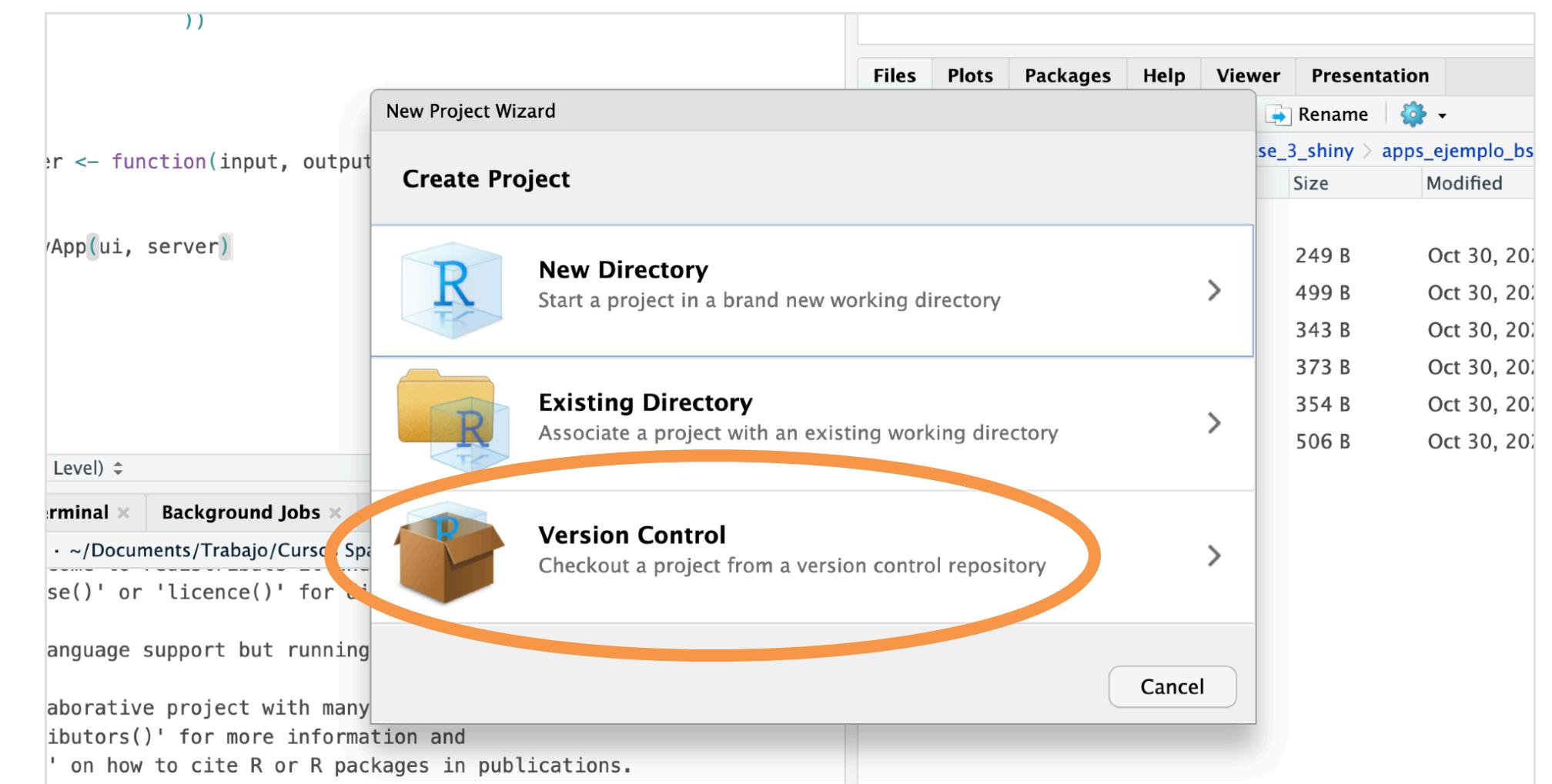
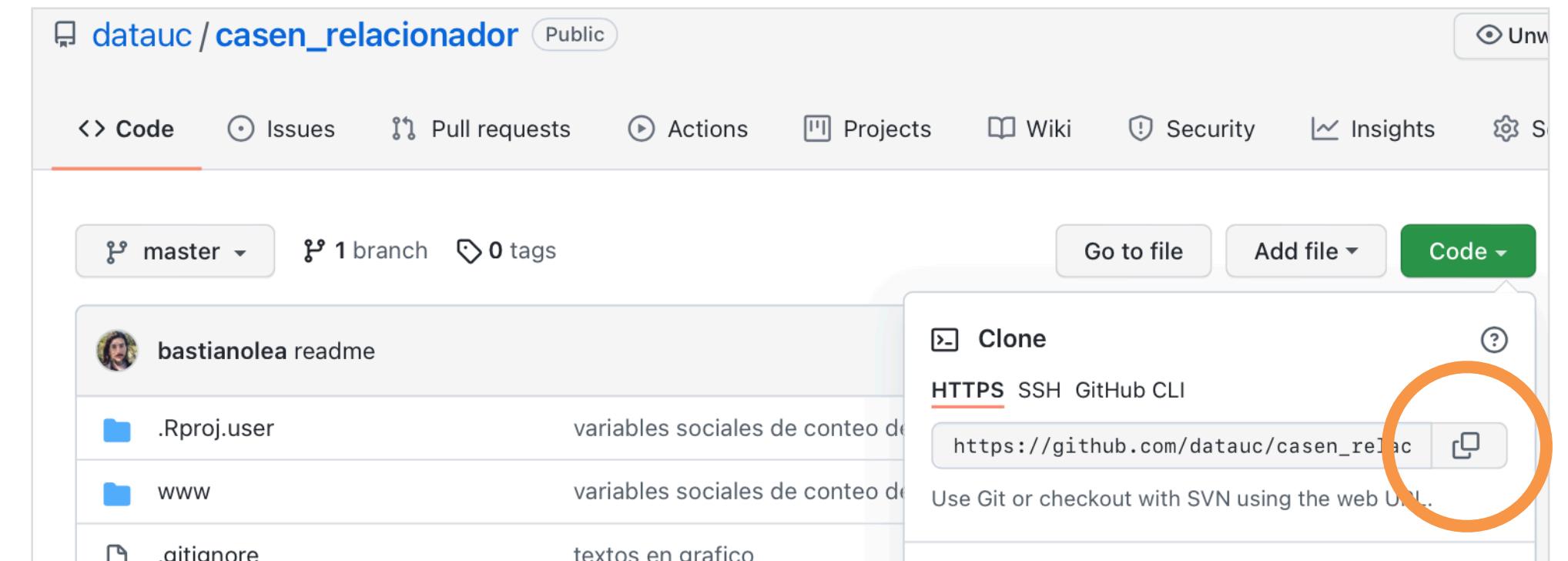
- Acceder a RStudio Server;
- En el navegador:
`direccionip:8787`



DigitalOcean

Instrucciones Digital Ocean

- Subir tu aplicación a GitHub, o encontrar una que quieras usar
- Copiar su enlace
- En RStudio, clonar un proyecto desde control de versiones (GitHub), o subirlo
- Probar la aplicación localmente



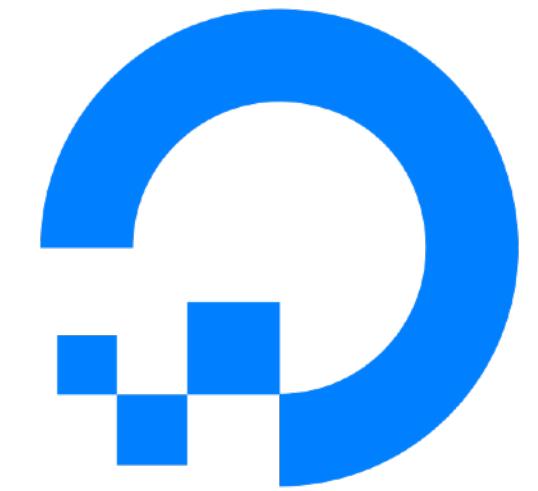
Instrucciones Digital Ocean

- Copiar la aplicación a la carpeta shiny-server:

```
sudo ln -s ~/miaplicacion /srv/shiny-server/
```

- Instalar paquetes para el usuario Shiny:

```
sudo su - shiny  
R  
install.packages("...")
```



Instrucciones Digital Ocean

- Editar configuración de Shiny:

```
sudo nano /etc/shiny-server/shiny-server.conf
```

Agregar lo siguiente:

```
preserve_logs true;  
sanitize_errors false;
```

(guardar usando ctrl + O, cerrar usando ctrl + W)

- Reiniciar shiny-server:

```
sudo systemctl restart shiny-server
```

Los errores podemos encontrarlos en la carpeta de logs:

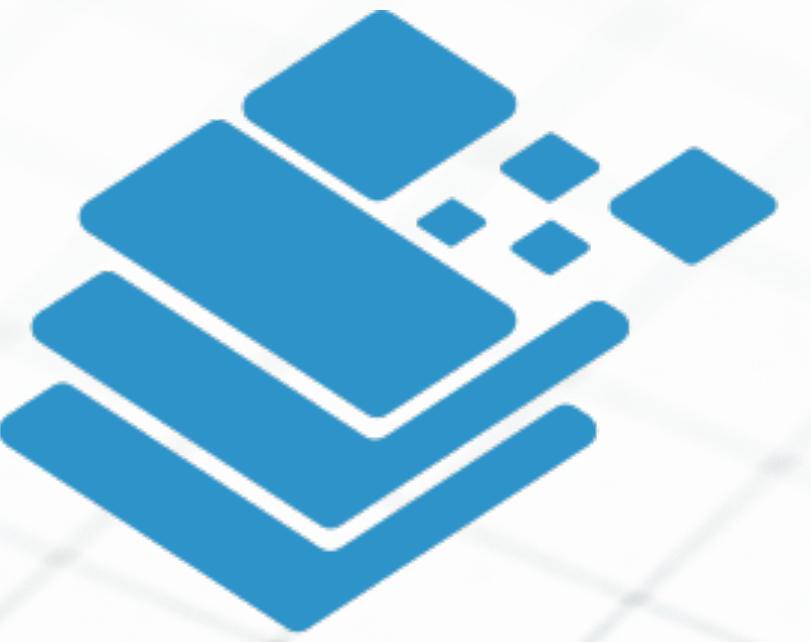
```
cd /var/log/shiny-server
```



Acceder a la aplicación

111.111.111.111:3838/nombredelaapp





**Spatial
Lab
Analytics**

Soluciones en análisis de datos

www.spatiallab.cl