Exercise V – Advanced Shiny

Part 1 – Deploy Shiny & RStudio Cloud

CEE412 / CET522

TRANSPORTATION DATA MANAGEMENT AND VISUALIZATION

WINTER 2020

Advanced Shiny Part 1

- In this exercise, we will introduce
  1. how to deploy your Shiny App on shinyapps.io
  2. how to use write and run R code online using RStudio Cloud. It is kind of an online version of RStudio to substitute your local RStudio software.

- The Shiny App you developed for the final project are required to deployed on shinyapps.io to help others easily access to your App.

Deploy Shiny App

- Firstly, check a data visualization demo (similar to the demo in Exercise 4 Part 4), which has been deployed on the Web: https://zhiyongc.shinyapps.io/sqlserver/
  - The Shiny apps deployed on **shinyapps.io** can be publicly accessed.
  - By using **shinyapps.io,** you can publish your final project, or make a Shiny App to demonstrate your research results online.

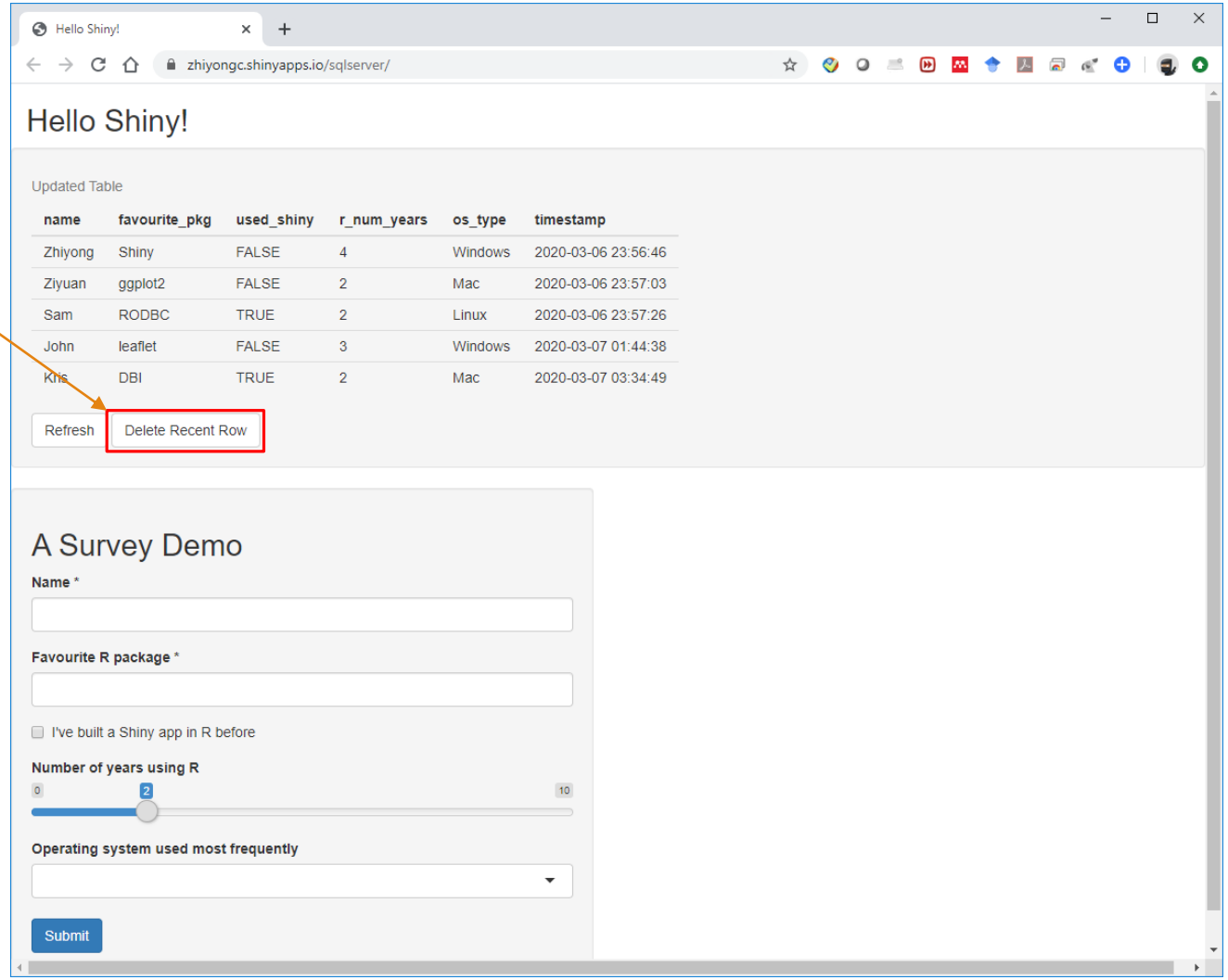
- Now, let's do it.

Step 1: Create a Local Shiny App

- We first create a local Shiny App
  - The functions in this Shiny App
    1. Connect to our SQL Server database to visualize the survey data
    2. Input survey data into the database
    3. Delete the more recent added record from the database and visualize the data
  - Review the UI of the deployed Shiny App in the next page

Step 1: Create a Local Shiny App

- A new function:
  - Deleting the most recent added row and updating the table



- Page link:

https://zhiyongc.shinyapps.io/sqlserver/

Step 1: Create a Local Shiny App

- Now, let's check out the source code
  - Exercises → Exercise 5 → Scripts → sqlserver → app.R
    - Note: we recommend you to put your script (app.R) into a new folder before uploading/deploying it

- New sections in the source code
  1. Use DBI and odbc to connect to database.

```
library(DBI)
library(odbc)
```

Install them before library them

```
# build connnection
conn <- DBI::dbConnect(odbc::odbc(),
                       Driver   = "SQLServer",
                       Server   = "128.95.29.72",
                       Database = "CEE412_CET522_W20",
                       UID      = "your username",
                       PWD      = "your password",
                       Port     = 1433)
```

Critical Note:

- Please user "SQLSever" as the Driver **when you deploying Shiny App**.
- But **when you run your code locally**, you may need to **add a space** between SQL and Server, like "SQL Sever".
- Please try both if you encounter database driver problems.

Change to your own database

Step 1: Create a Local Shiny App

- New sections in the source code
  2. In UI, add a new button

```
column(12,
          wellPanel(
            tableOutput('table'),
            actionButton(inputId = "refresh", label = "Refresh"),
            actionButton(inputId = "delete", label = "Delete Recent Row")
          )
    )
```

Step 1: Create a Local Shiny App

- New sections in the source code
  3. In server function, create a list of reactiveValues (rv) to store the data to be displayed in the table. The data to store query results is named as tableData

```
# initialize the tableData as the query result that stored in initial_surveyData
  rv <- reactiveValues(tableData = initial_surveyData)

  output$table <- renderTable(rv$tableData,
                              caption = "Updated Table",
                              caption.placement = getOption("xtable.caption.placement", "top")
  )
```

Step 1: Create a Local Shiny App

- New sections in the source code
  4. In server function, add an observeEvent to response to the clicking of the delete button
     - We add a query to delete the most recent added row and execute it using dbSendQuery()
     - Then, we do the same thing to query data from E4_Survey to update rv$tableData

```
observeEvent(input$delete, {
  # Query to delete the most recent added row
  deleteQuery <- "DELETE [E4_Survey] WHERE [timestamp] = (SELECT TOP 1 [timestamp] FROM [E4_Survey] ORDER BY [timestamp] DESC)"
  # execute delete query
  dbSendQuery(conn, deleteQuery)

  # Select all data from E4_Servery
  surveyQuery <- "SELECT * FROM [E4_Survey]"
  surveyData <- dbGetQuery(conn, surveyQuery)
  surveyData$timestamp <- as.character(as.POSIXct(surveyData$timestamp, origin="1970-01-01", format="%d/%m/%Y %H:%M:%S"))
  rv$tableData <- surveyData
})
```

Step 1: Create a Local Shiny App

- Please note:
  1. Since you cannot write data into the E4_Survey table in the CEE412_CET522_W20 database, you can import the E4_Survey into your own database, as you did before.

  2. The SQL query execution in DBI & ODBC packages is different from that in RODBC package.
     - Since you are required to publish your Shiny App. **You need to start to use DBI & ODBC packages**.
     - Document of Shiny's database access with ODBC
       - https://docs.rstudio.com/shinyapps.io/applications.html#accessing-databases-with-odbc
     - Document of the DBI package
       - https://cran.r-project.org/web/packages/DBI/DBI.pdf

  3. If you have any questions about executing your queries in R, please feel free to post it on Piazza or let the instructor/TA know.

Step 1: Create a Local Shiny App

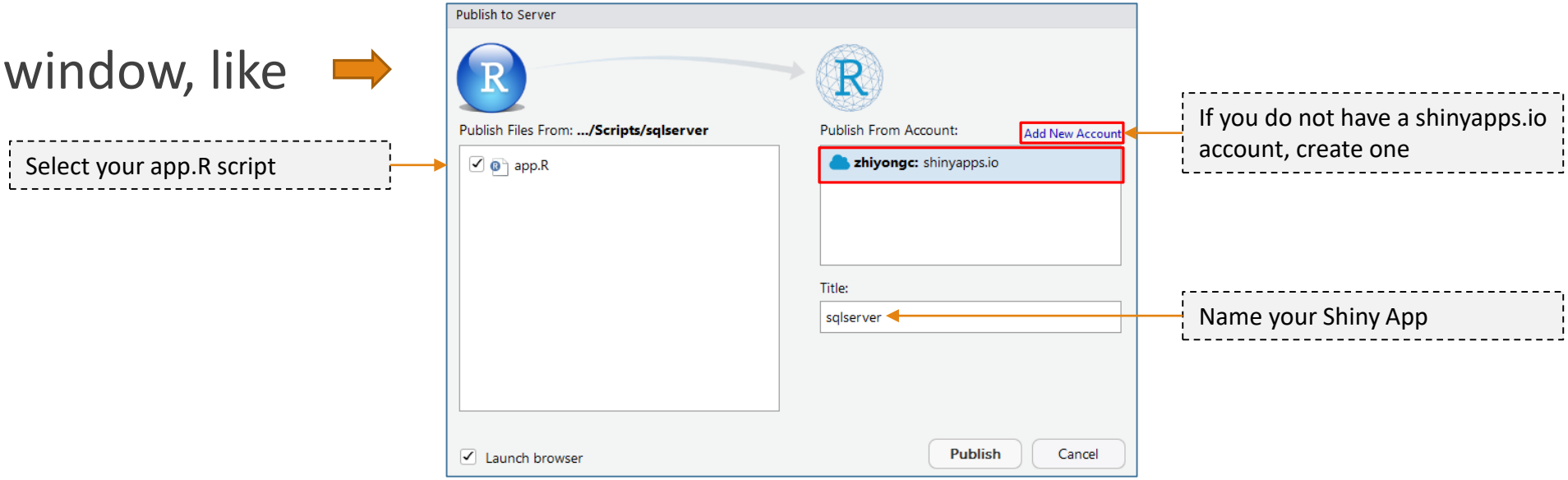
- Now, try to run the code in your local computer to ensure there is no error.
  - Remember to adjust the database driver, if needed.

```
# build connection
conn <- DBI::dbConnect(odbc::odbc(),
                       Driver   = "SQLServer",
                       …)
```

- If you can run the code successfully, find the 

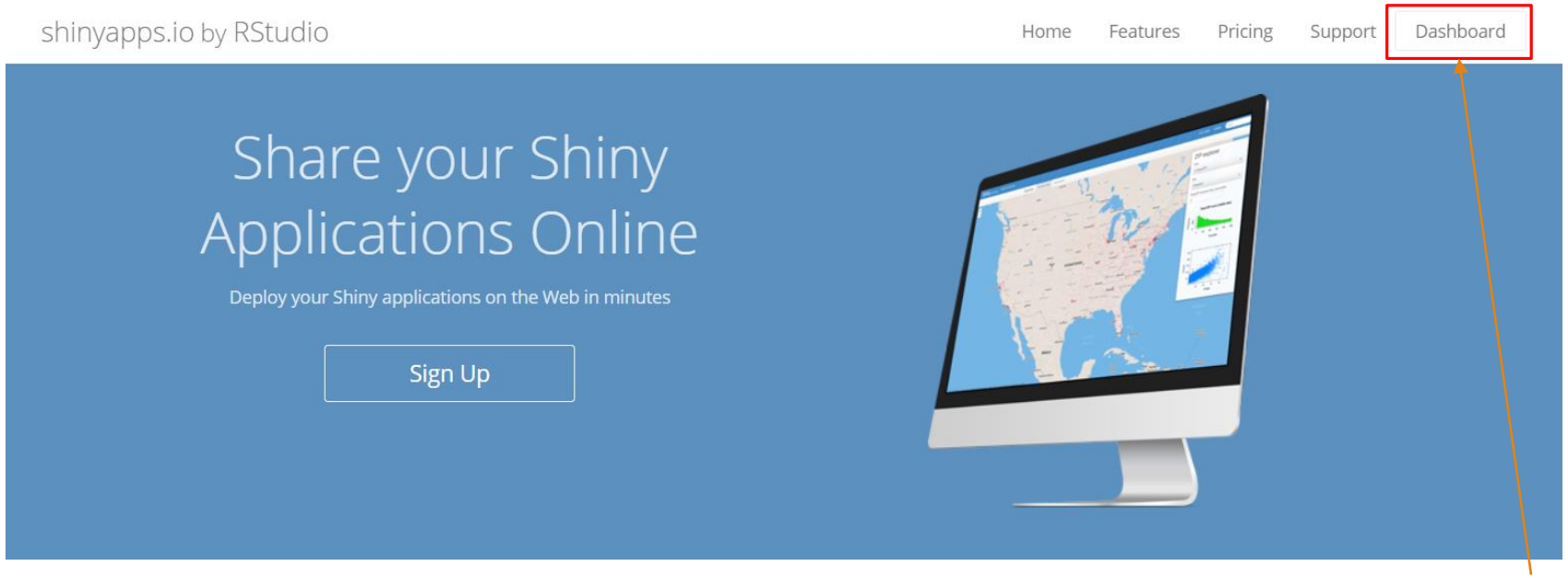
icon at the right top corner of your Shiny App interface and click on it.

- You will get a window, like



Step 2: Create a shinyapps.io Account

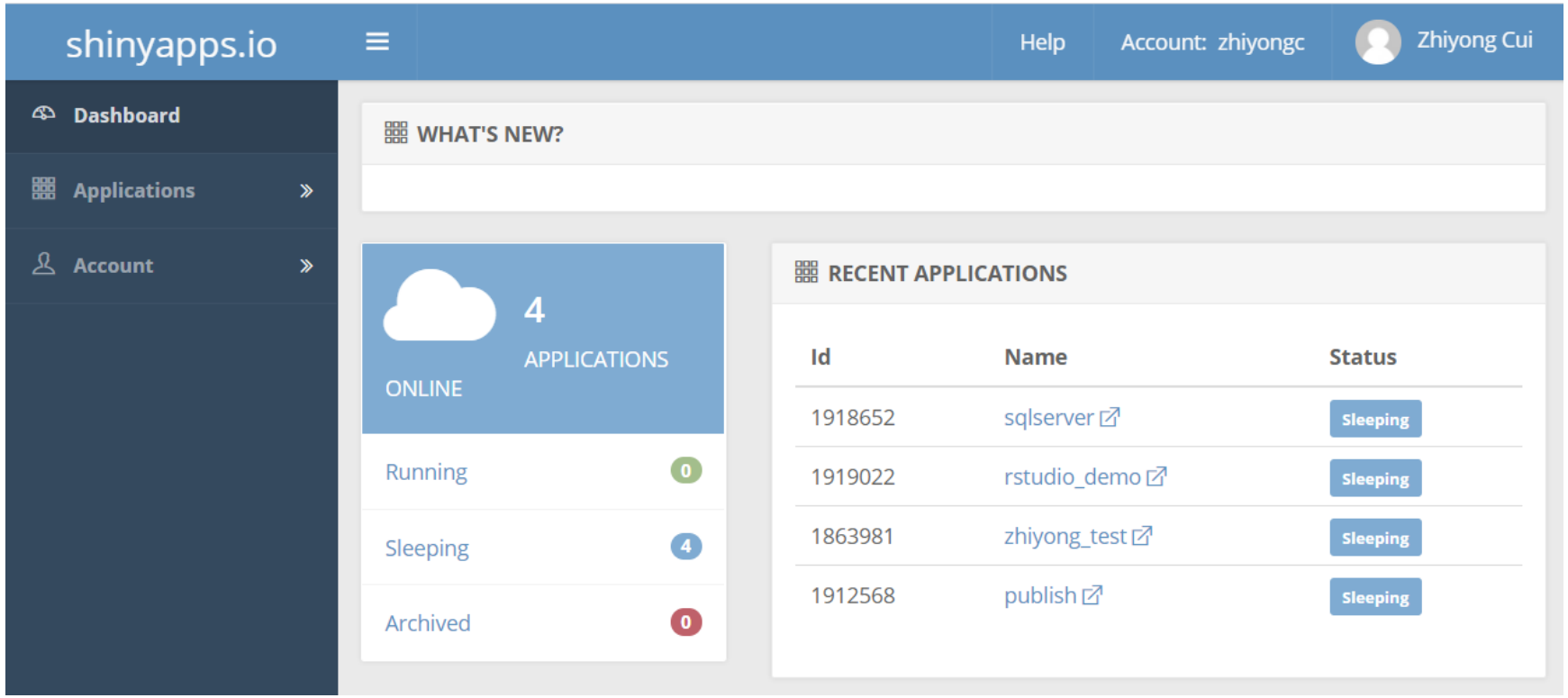
- Create a shinapps.io account



- Then, login to your shinapps.io account by clicking the Dashboard button
- You can check all your shiny apps deployed on shinapps.io
  - Your free account can have at most 5 apps deployed.
  - Check more info in document: https://docs.rstudio.com/shinyapps.io/

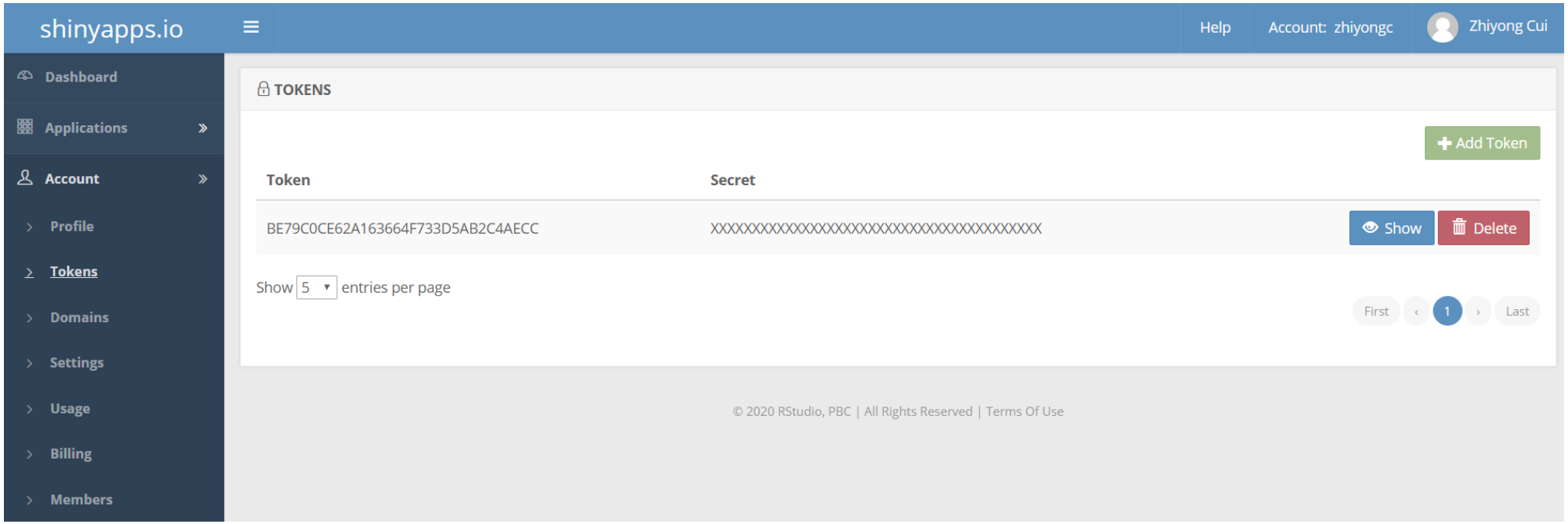
Step 2: Create a shinyapps.io Account

- Shinyapps.io Dashboard example:
  - I have for applications online. You view more details by clicking them.



Step 2: Create a shinyapps.io Account

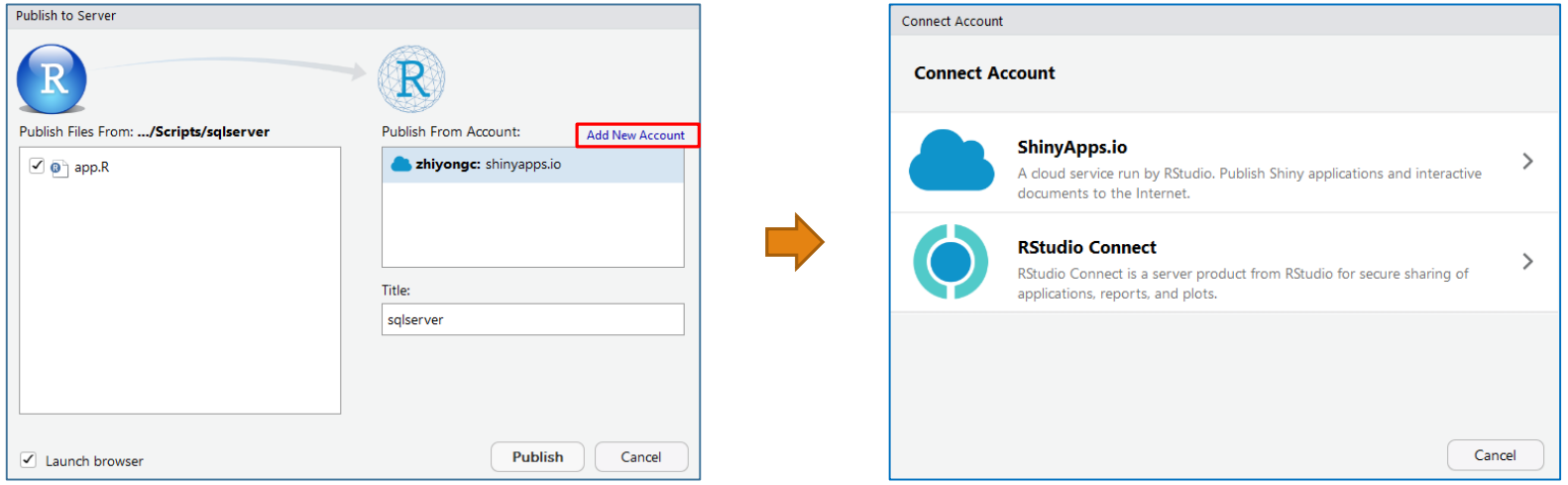
- In the left menu → Account → Tokens, the token is used to configure rsconnect package to access your account when you deploying your apps.



- Please check more details about deploying apps in this document:
  - https://docs.rstudio.com/shinyapps.io/getting-started.html#deploying-applications

Step 3: Publish

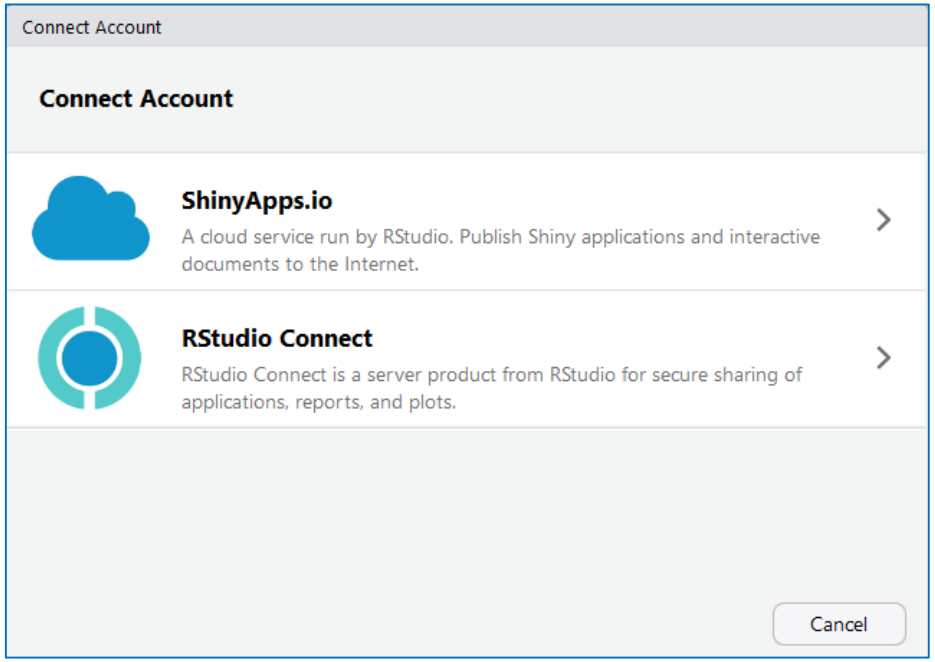
- After you get a shinyapps.io account, connect your account

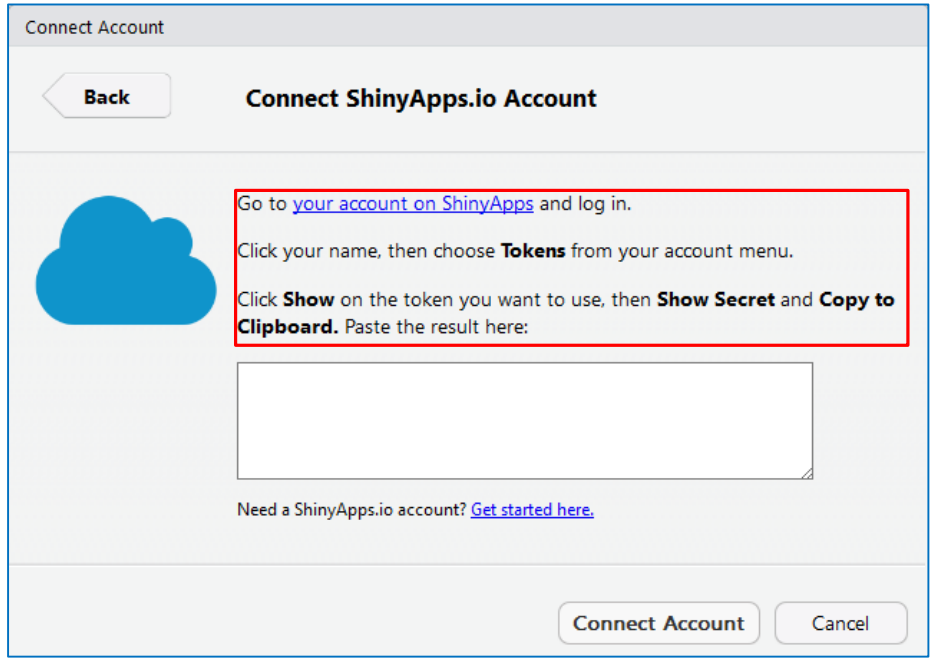


- Choose ShinyApps.io.
  - You can also choose RStudio Connect if you have a RStudio Cloud account. We will show you how to use RStudio Cloud later.

Step 3: Publish

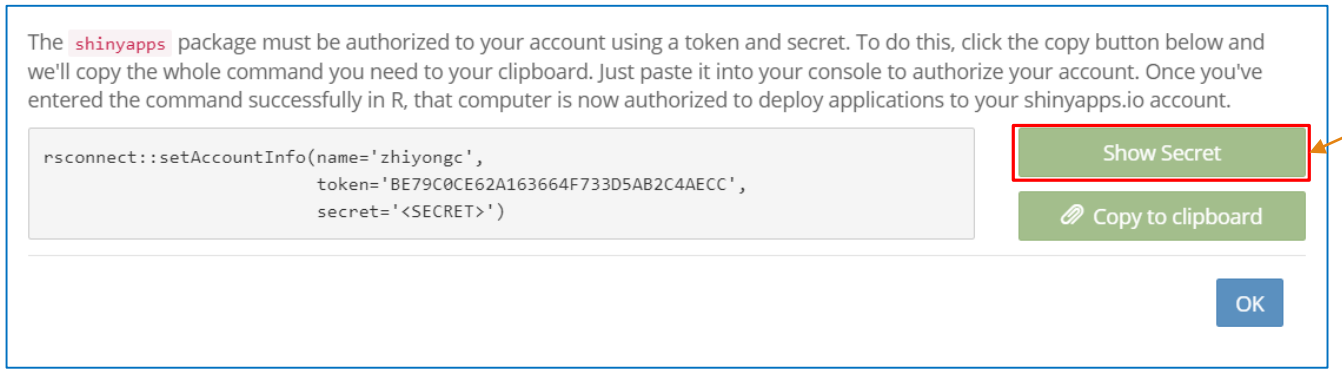
- Choose ShinyApps.io



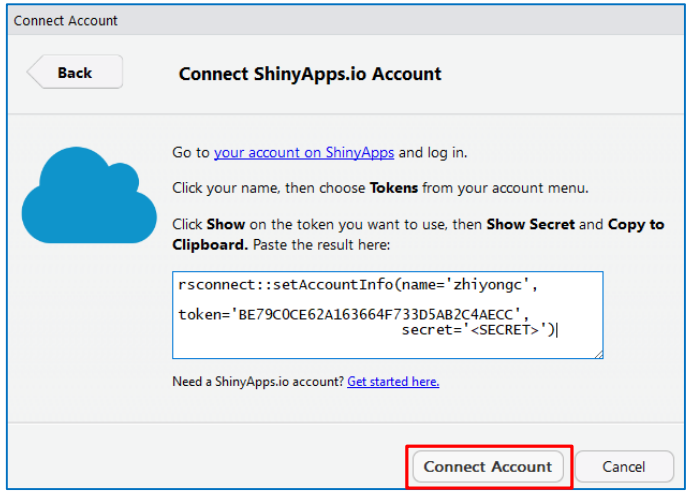


Step 3: Publish

- Get your Tokens from your **ShinyApps.io dashboard** according to the instruction.
  - When copying token information, please remember to click Show Secret:

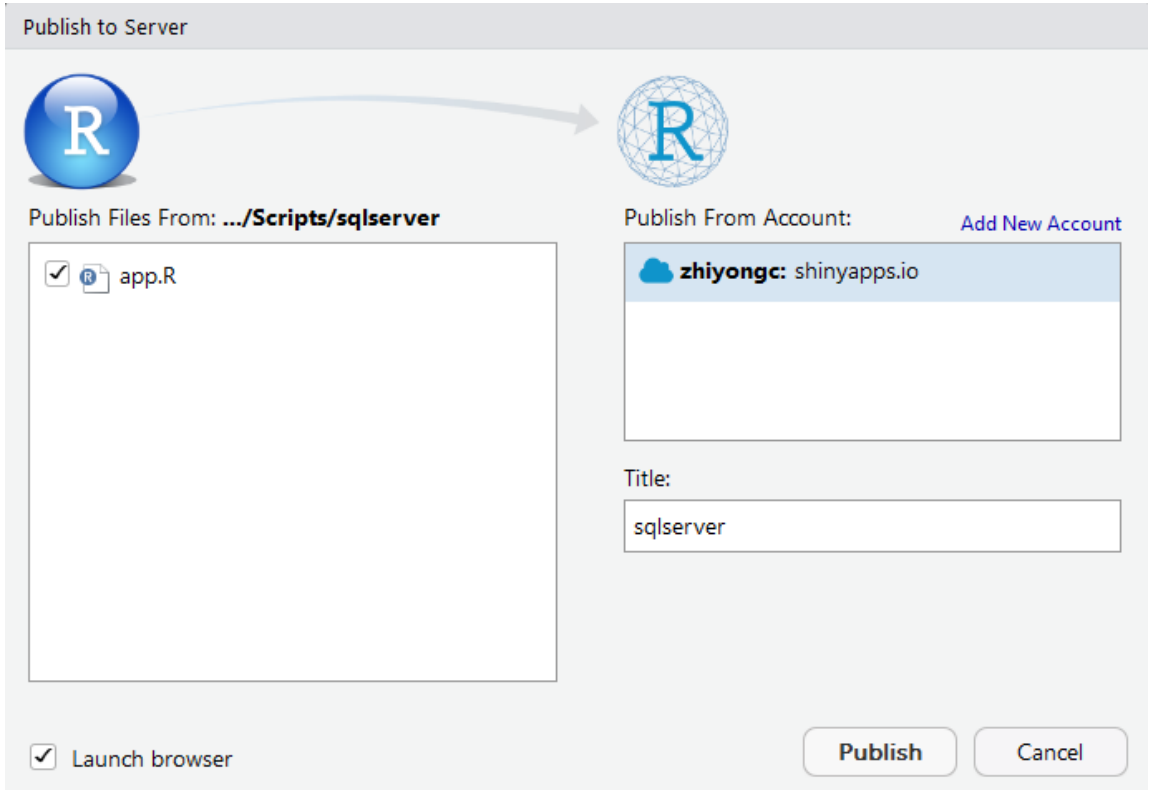


- After pasting your tokens, click **Connect Account**



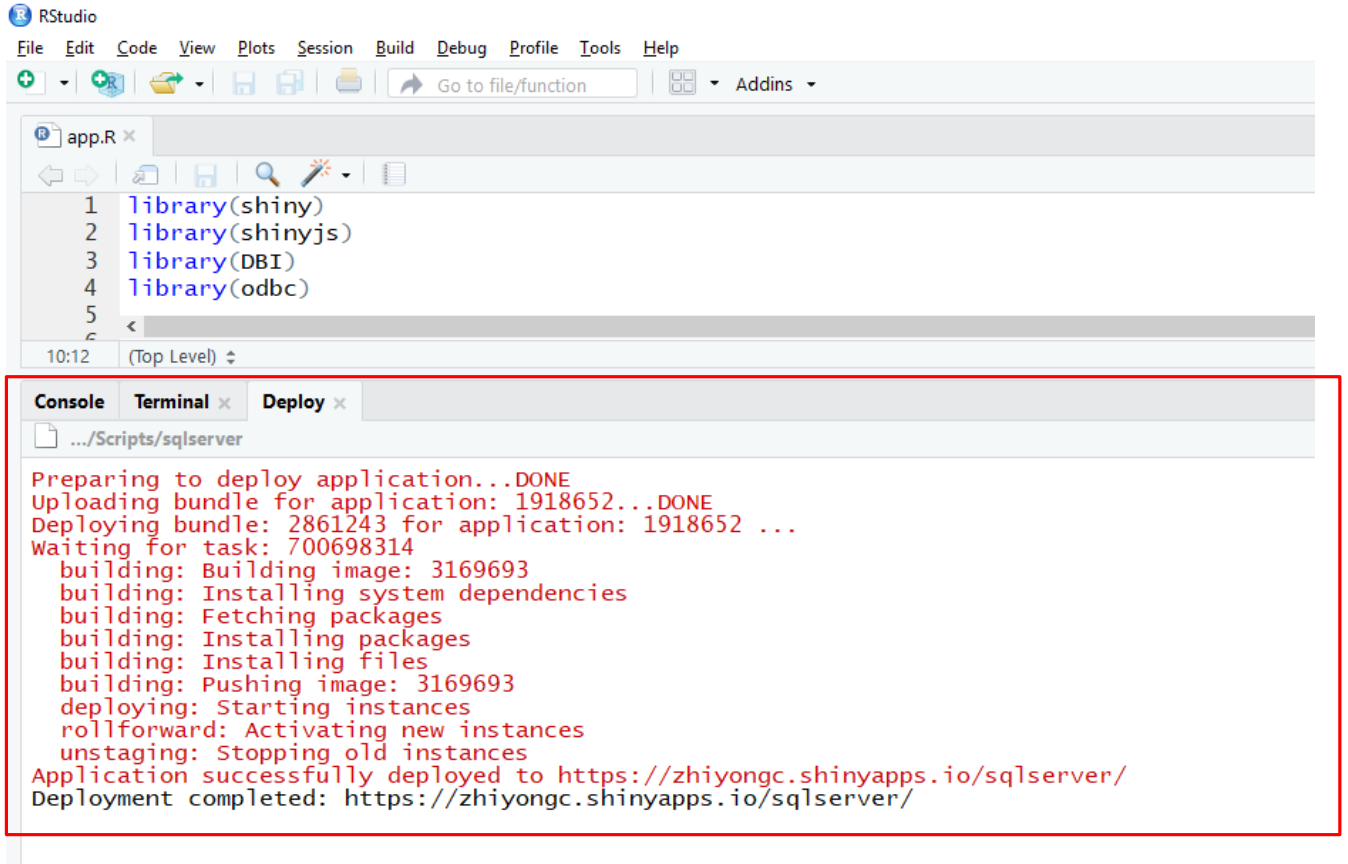
Step 3: Publish

- Then, publish your app by clicking Publish



Step 3: Publish

- A Deploy Panel at the bottom of RSudio Console will show up. The deploy process will be shown in this panel:



Step 3: Publish

- After the publishing process finished, a webpage will show up in your browser.
  - My example: https://zhiyongc.shinyapps.io/sqlserver/
- The website should be identical as the one we show you at the beginning of this exercise.
- You can also find your app in the ShinyApps.io Dashboard Now.
- Remember this website URL. You can share it and it is publicly accessible.
- You can play with this demo and adjust the code.
- Then try to re-publish/re-deploy your Shiny App.

ShinyApps.io

- In summary, deploying your Shiny App is pretty easy and useful, especially you need to share your App as a website.

- Find more user guides of ShinyApps.io here:
  - https://docs.rstudio.com/shinyapps.io/

- One thing you need to know: if you want to use ShinyApps.io to access to your own local databases or cloud databases, such as AWS databases, you may need to configure the firewall of your computers or AWS.
  - In this class, we have already set them up. Thus, you don't need to configure that for the class database.

RStudio Cloud

RStudio Cloud

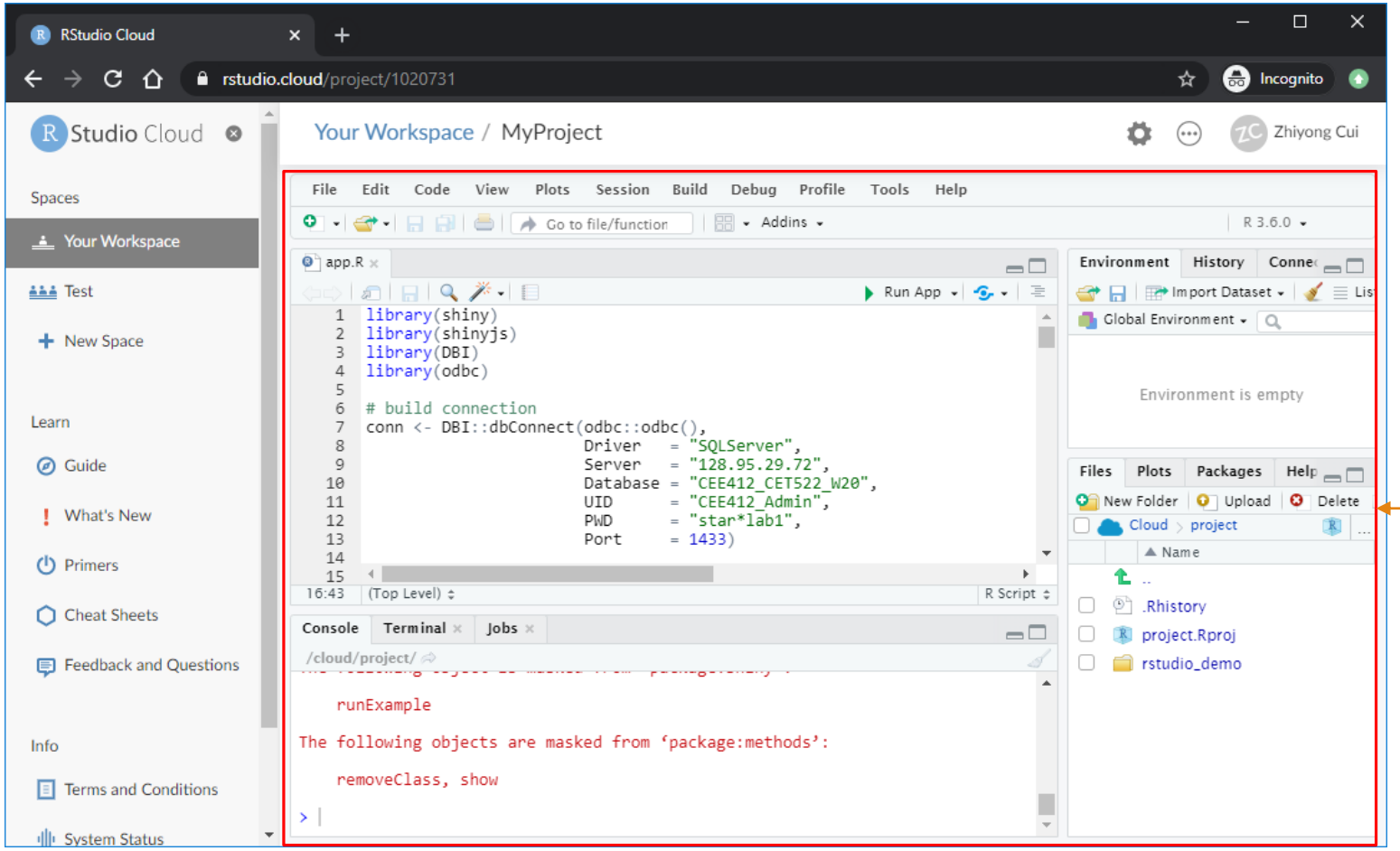
- RStudio Cloud (https://rstudio.cloud/) is a cloud environment that you can use R to do, share, teach and learn data science.

- RStudio Cloud provide you with workspace, in which you can create your projects. In each project, you can create new R scripts, Shiny Apps, and other R-enabled functions. You can also install whatever packages you need.

- It also provide you a virtual RStudio environment. That means you can do all the previous R-related or Shiny-related Exercises on RStudio Cloud.

RStudio Cloud

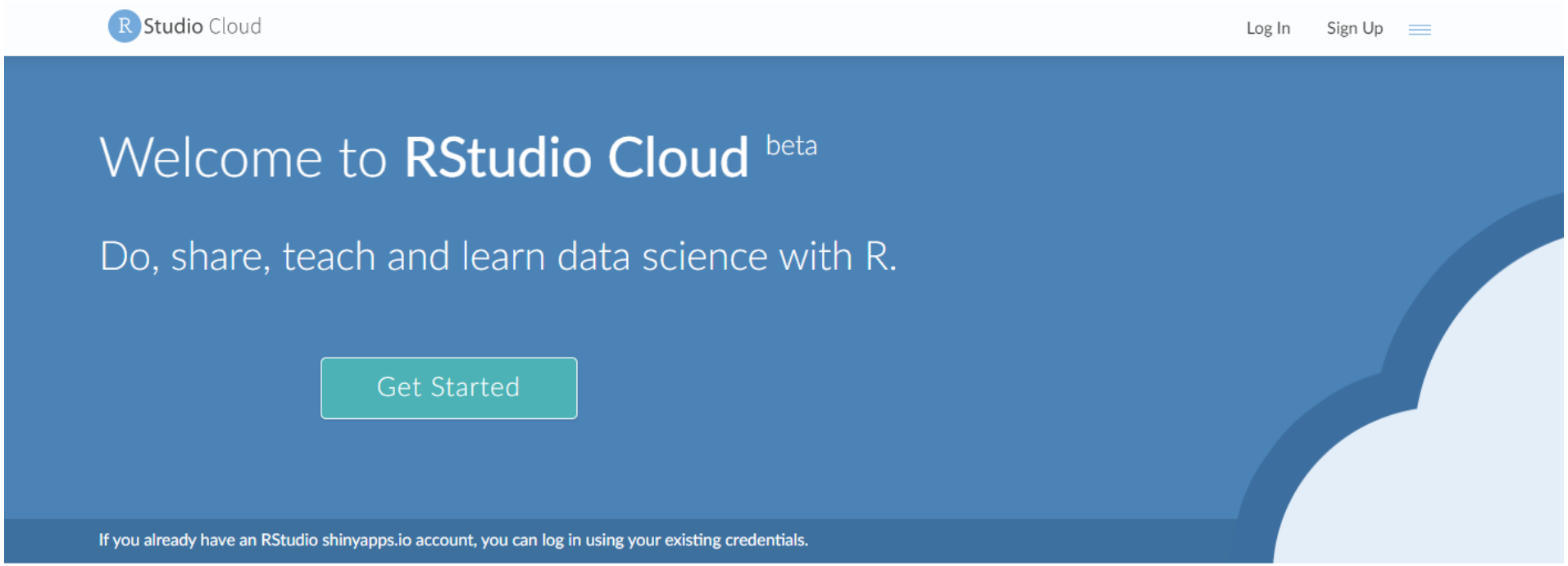
- A demo of the RStudio Cloud UI:



This area is exactly same as your local RStudio UI

RStudio Cloud

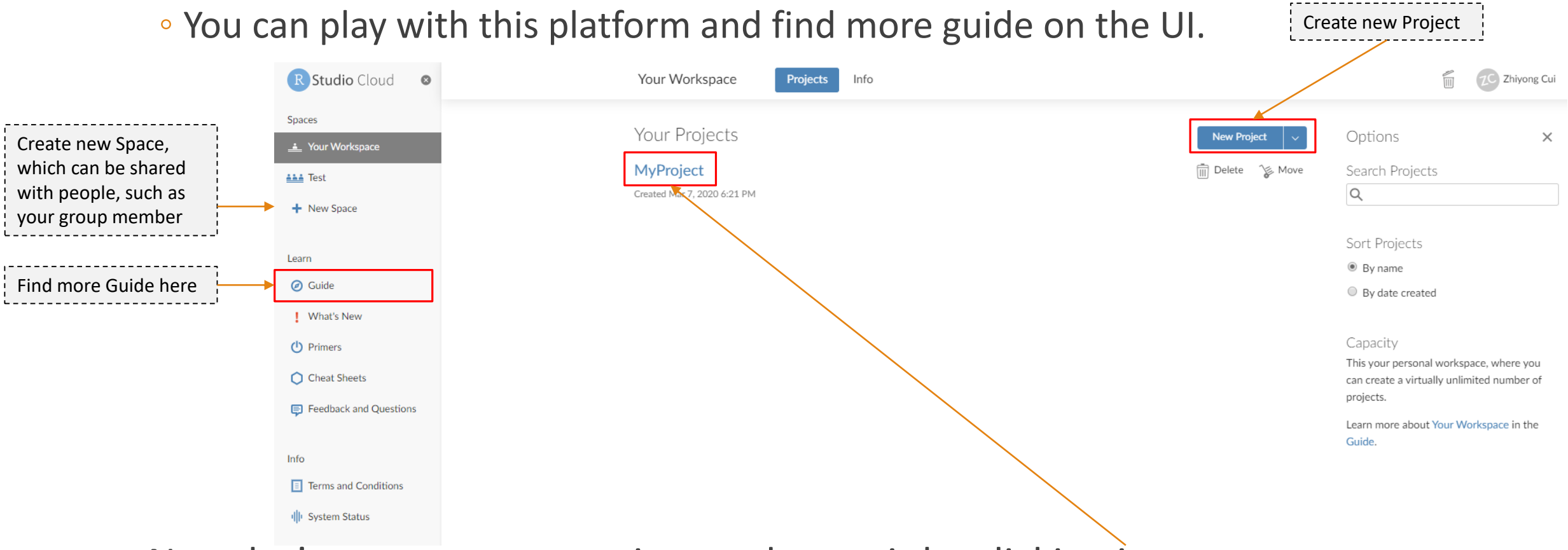
- Now, Get Started: Sign Up → Log In



- If you already have an RStudio shinyapps.io account, you can log in using your existing credentials.

RStudio Cloud

- After log in, you can find a UI like the following figure:
  - You can play with this platform and find more guide on the UI.

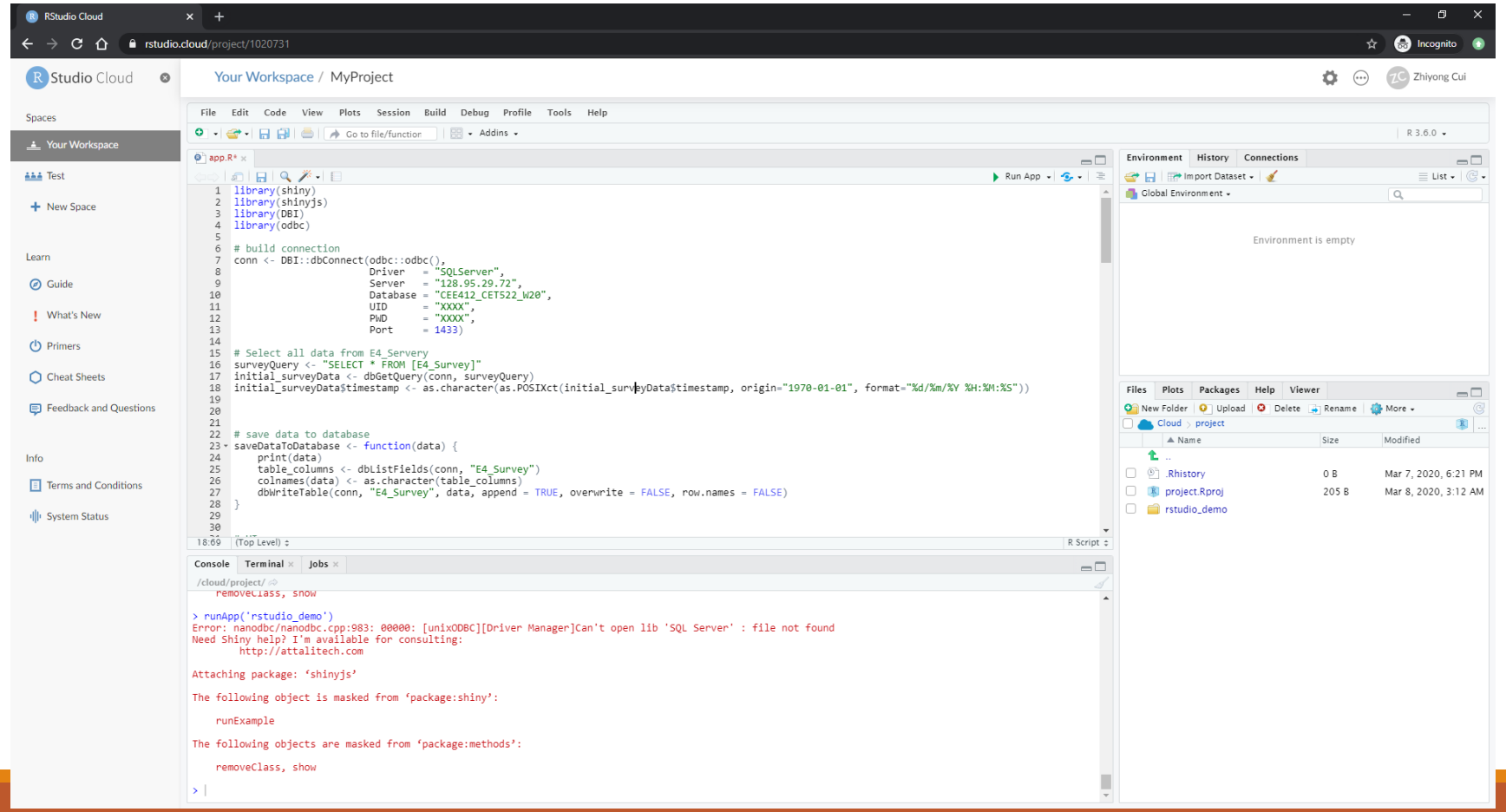


- Now, let's create a new project and open it by clicking it

RStudio Cloud

- Create a app.R script. Copy and paste our demo code into it.
  - Demo source code: Exercises → Exercise 5 → Scripts → sqlserver → app.R

Remember to install packages and configure your database connection first.



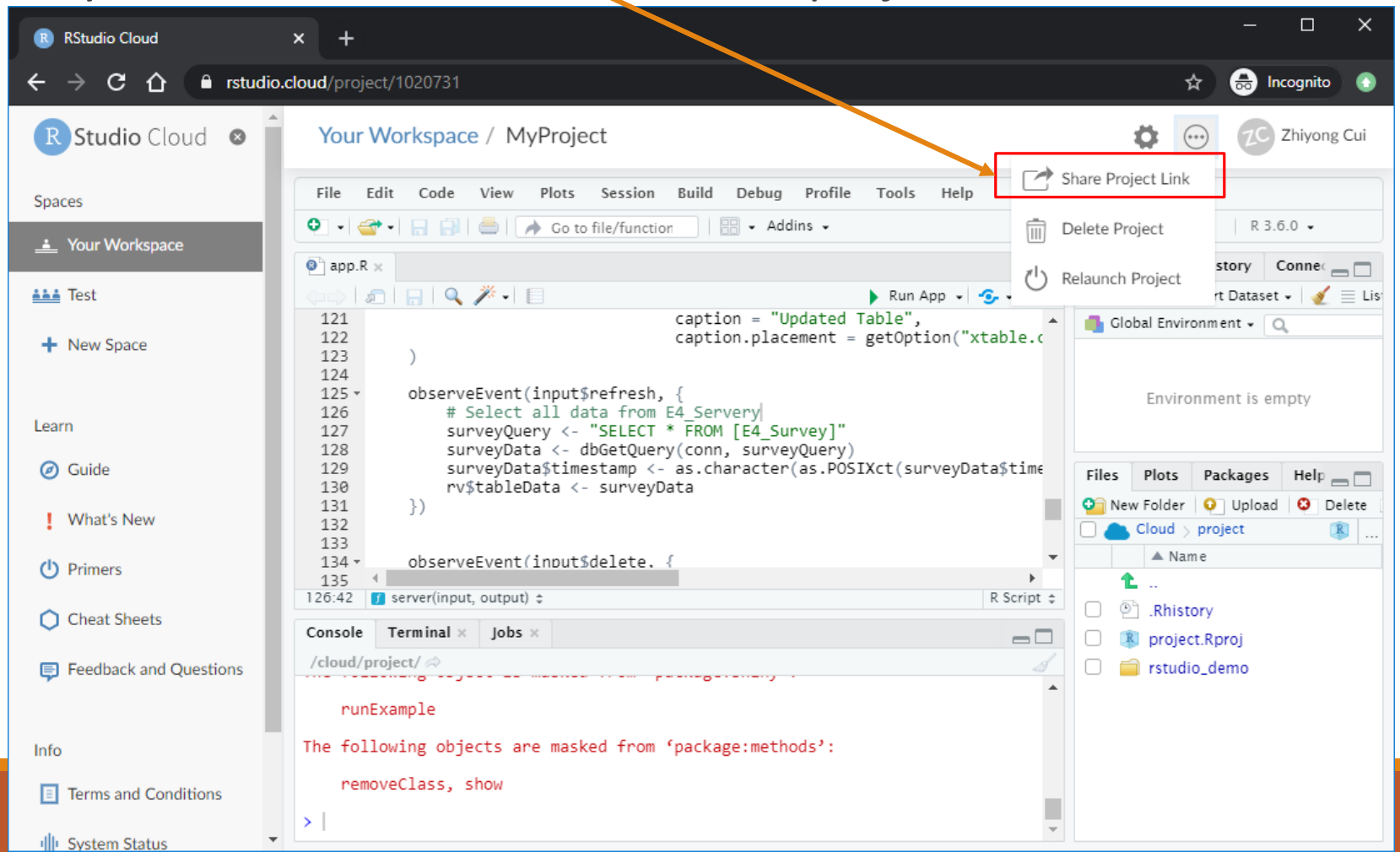
RStudio Cloud

- Then, try to Run your Shiny App via the RStudio Cloud.

- Next, try to re-publish/re-deploy your Shiny App via the RStudio Cloud.

RStudio Cloud

- You can even share your project, which is good for your group projects.
  - But only one user can access the shared project at the same time.



RStudio Cloud

- If you have difficulty to use R or RStudio on your local computers (for example, computers with Mac or Linux), RStudio Cloud is a good choice.

- It is pretty fast and free. Your code will be stored on cloud, which will not be lost.

Summary

- For your final project, you are required to **publish your Shiny App before the final presentation**.

- If you want to participate Shiny Contest 2020 https://blog.rstudio.com/2020/02/12/shiny-contest-2020-is-here/, please note that they require the participants to use ShinyApps.io and RStudio Cloud.

- Next, we will practice more to make your Shiny App more powerful and beautiful.