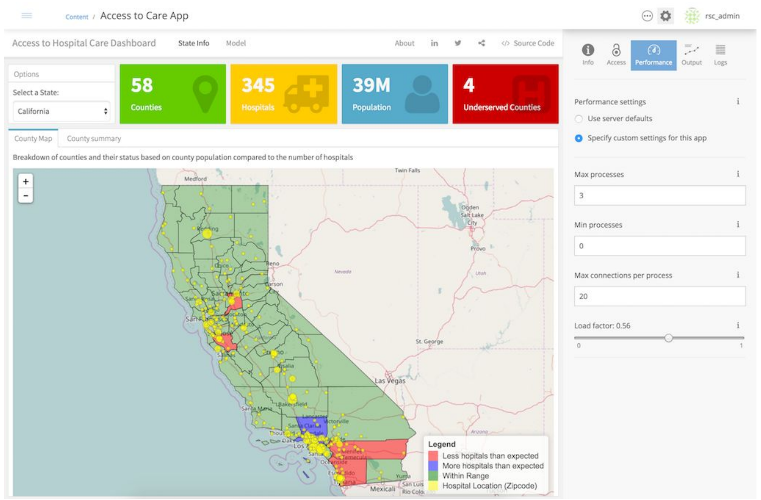
RStudio Connect in Production

September 2019

Thomas Mock



Select a State: California

Breakdown of counties and their status based on county population compared to the number of hospitals

🟥	Less hopitals than expected
🟦	More hospitals than expected
🟩	Within Range
🟨	Hospital Location (Zipcode)

Performance settings

Use server defaults [✓ Specify custom settings for this app]

Max processes: 3

Min processes: 0

Max connections per process: 20

Load factor: 0.56

RStudio

Today's Agenda

- Overview of Connect
- Overview of R Markdown on Connect
- Demo!

Additional topics worth exploring

- Best Practices for Administering R Studio in Production
- Model Management with RStudio Connect
- Shiny in Production: Principles, practices, and tools
- Rstudio Team Quickstart (Try Connect without installing anything)!
- solutions.rstudio.com

RStudio Connect

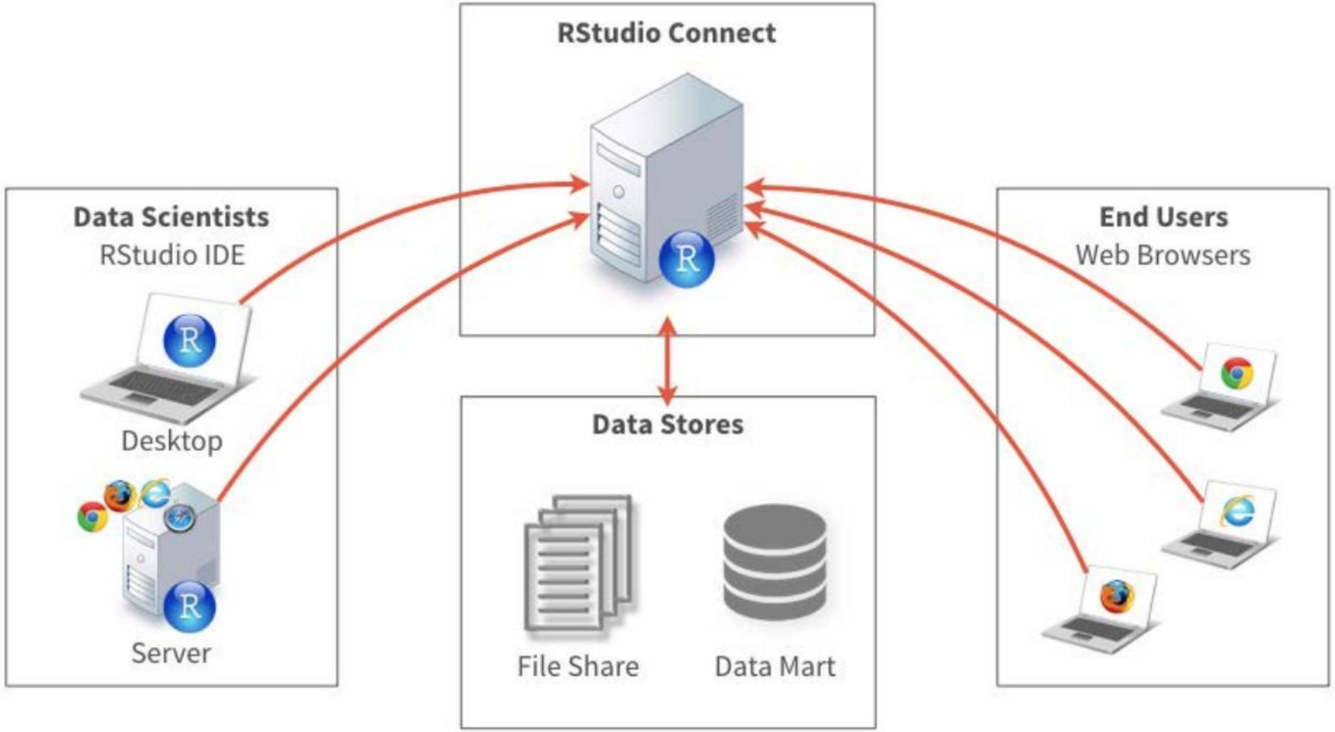
A content hub and execution engine for all your data products.

Makes R Insights Easy to **Leverage**

- Web Apps
- Machine learning
- RESTful APIs
- Dashboards
- Reports
- Emails
- Tables
- Visualizations

Fully **Interoperable**

- R
- Python
- Spark
- Javascript
- Tensorflow
- SQL


RStudio Connect
Data Scientists
RStudio IDE
Desktop
Server
Data Stores
File Share
Data Mart
End Users
Web Browsers

RStudio

Production

Production

What does production mean?

- Software environments that are **used** and **relied on** by real users, with real consequences if things go wrong - Joe Cheng

Production Needs

- Scaling and Execution
- Security and Access Control
- Logging/Visibility
- Reproducibility
- Sharing and Collaboration

Production doesn’t have to be scary

Connect

What does Connect provide?

- Scaling, Execution, and High Availability
- Security, Access Control, Visibility
- Reproducibility and Logging
- Sharing and Collaboration

Connect gets your data products **off your laptop** and ready for **consumption**!

Security

- Server Level (AD/LDAP/SAML/etc)
- App Level (User Permissions)
- Data Level (User Groups)

Reproducibility

Packrat Bundles

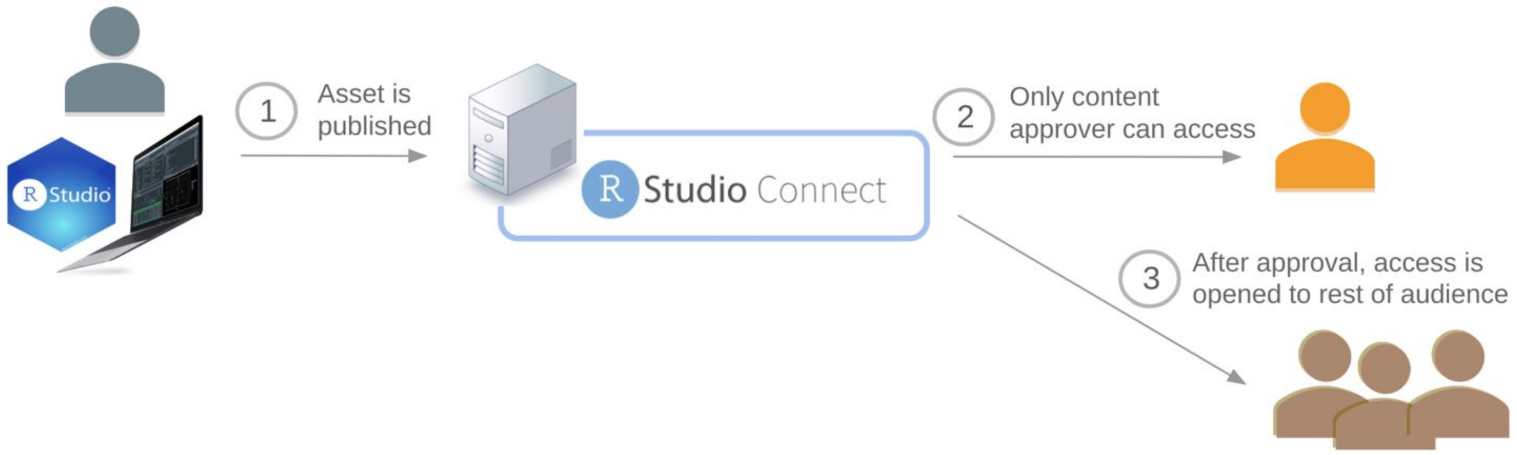
- Connect receives bundle identifying source code and specific dependencies
- Connect unpacks bundle and uses packrat to **install** the specific dependencies

- Each data product has own R environment, prevents breaking existing, deployed content

https://docs.rstudio.com/connect/admin/package-management.html

Deployment

Push-button deployment

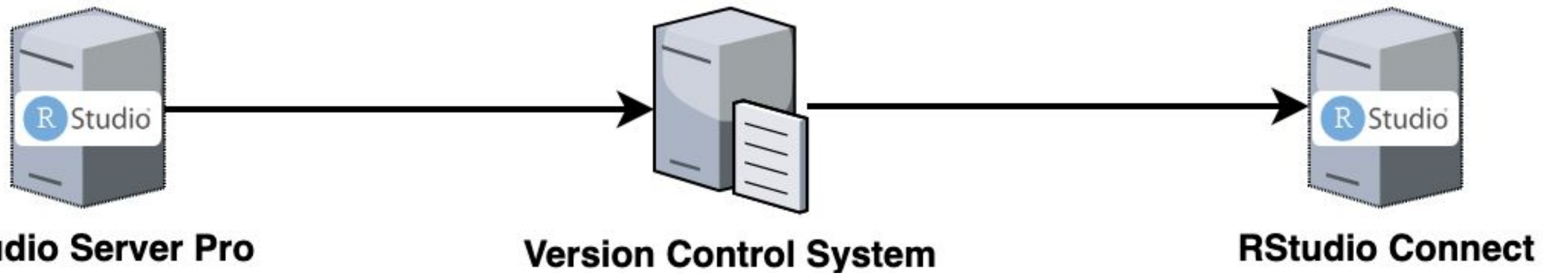

1
Asset is published
RStudio
RStudio Connect
2
Only content approver can access
3
After approval, access is opened to rest of audience

git-backed deployments

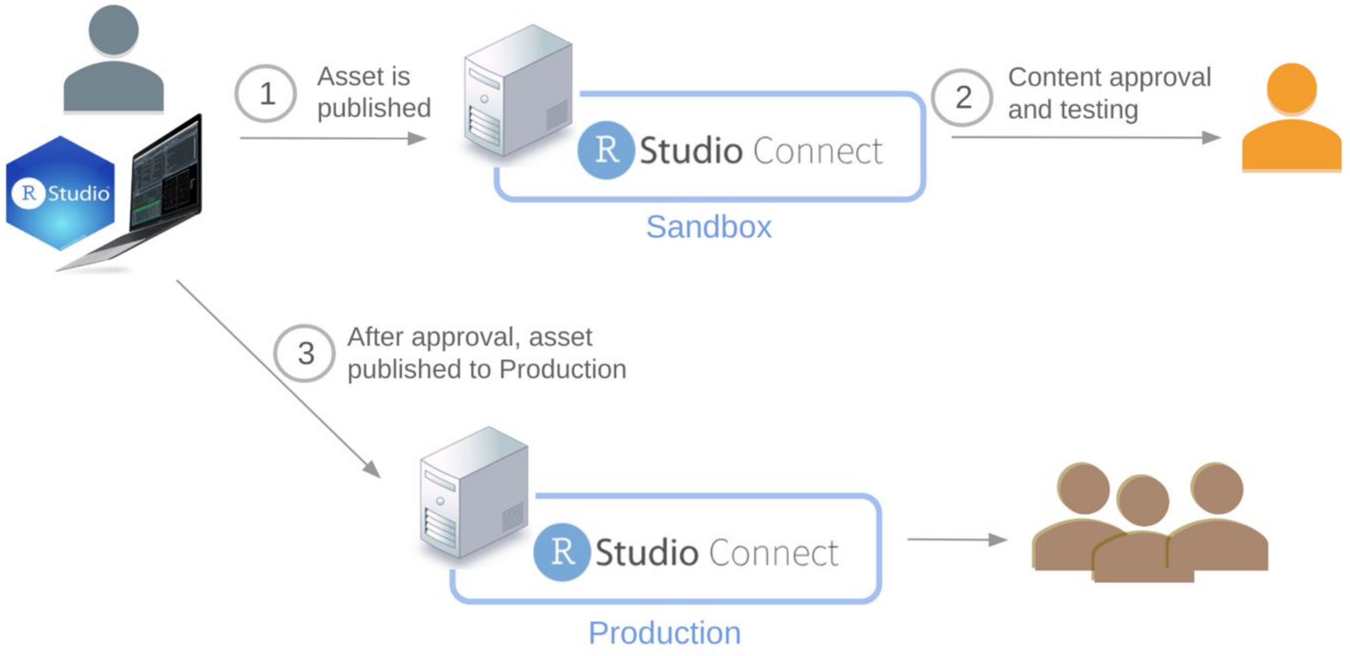
1) Configure Git repository in RStudio Connect

2) RStudio Connect publishes initial version

3) Updates are handled by RStudio Connect

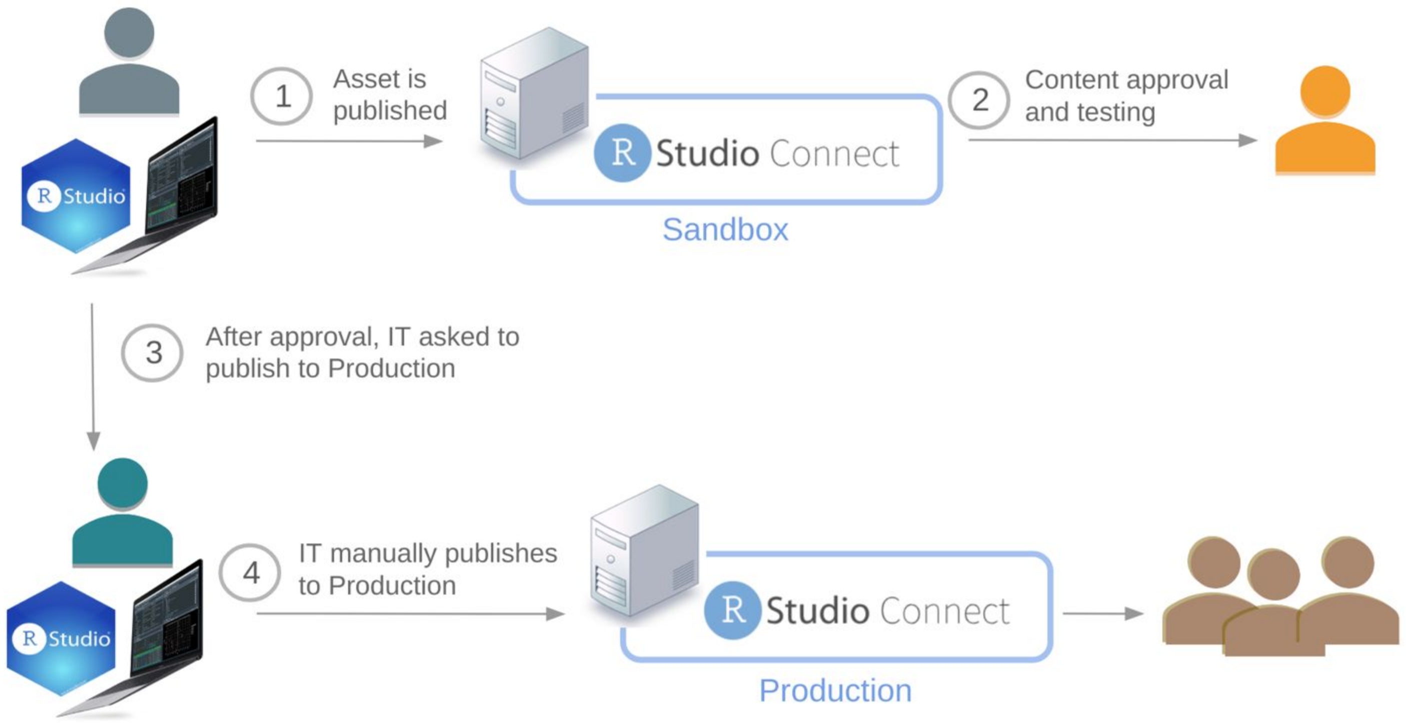


Test and Production

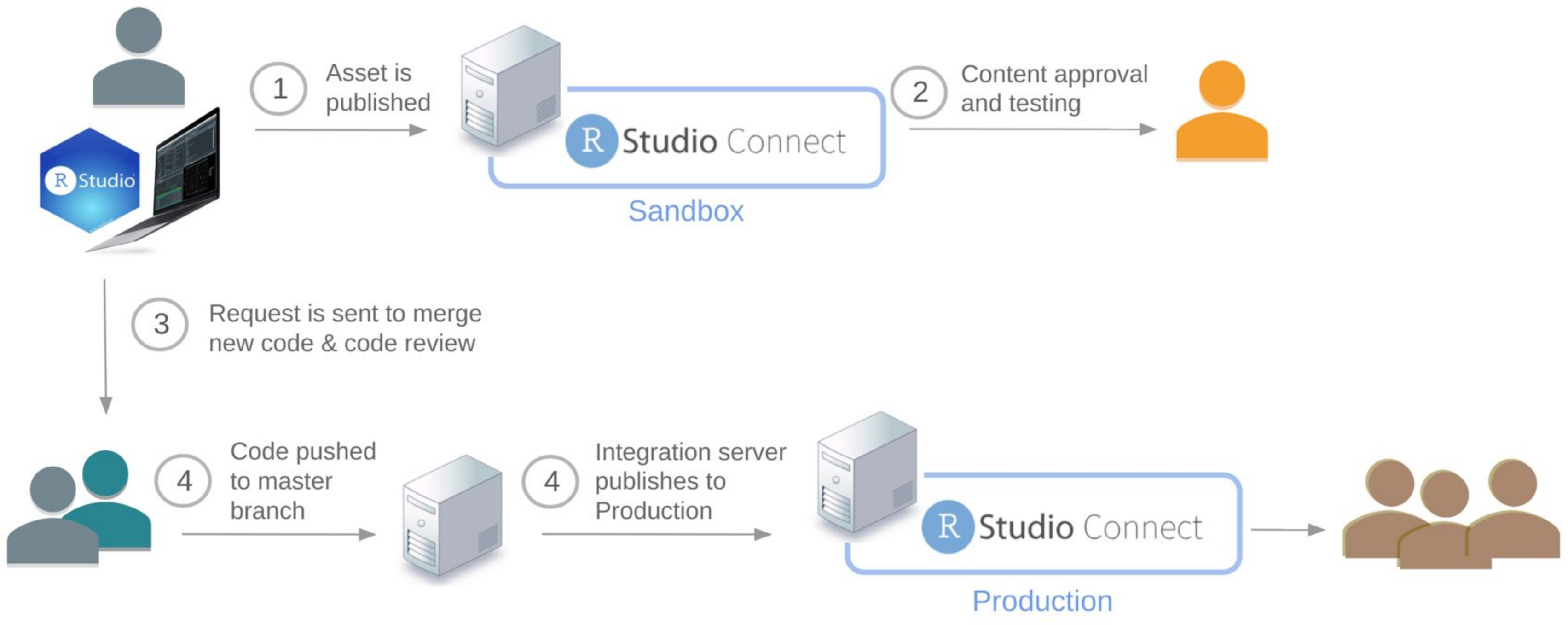


1
Asset is published
RStudio Connect
Sandbox
2
Content approval and testing
3
After approval, asset published to Production
RStudio Connect
Production

Request deployment


1
Asset is published
Studio Connect
Sandbox
2
Content approval and testing
3
After approval, IT asked to publish to Production
4
IT manually publishes to Production
Studio Connect
Production

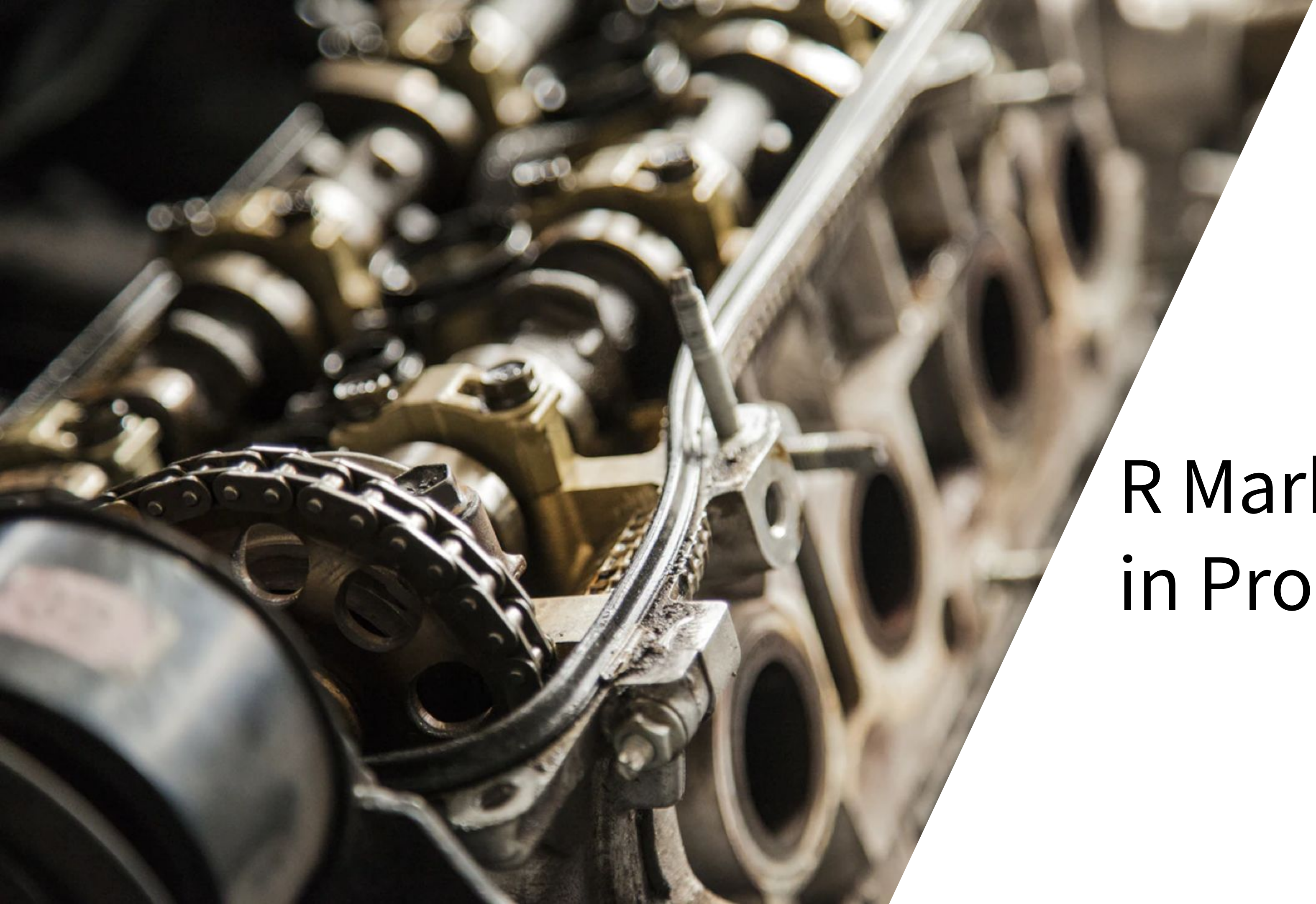
Continuous Integration


1
Asset is published
RStudio
RStudio Connect
Sandbox
2
Content approval and testing
3
Request is sent to merge new code & code review
4
Code pushed to master branch
4
Integration server publishes to Production
RStudio Connect
Production

Data Products

MORE INTERACTIVE ↓

- .Rmd
- .Rmd with parameters
- .Rmd with htmlwidgets
- .Rmd with Shiny components
- **Shiny app**



R Markdown in Production

R Markdown Reports

- Open source R package that produces high production quality documents by integrating prose with code and results.
- Targets data scientists/analysts with R expertise who want to share their analyses in a reproducible fashion
- Supports multiple output formats including:
  - Documents : PDF, HTML, Word, RTF, Markdown
  - Journals
  - Presentations
  - Dashboards
  - Websites
  - Books
- Authoring format for data science. Visit http://rmarkdown.rstudio.com/ to learn more

Parameterized Reports

Write the function

```r
render_impact_report <- function(states, years){
  rmarkdown::render(
    "param-report-programmatic.rmd",
    params = list(
      states = states,
      years = years
    ),
    output_file =
      glue::glue("animal-impact-{states}-{years}.html")
  )
}
```

Then specify the paramater inputs

```r
states <- rep("TX", 4)
years <- 2015:2018
```

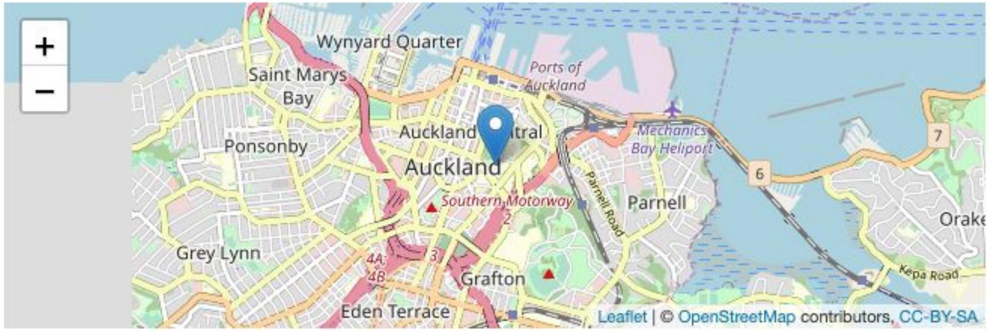
Then `knit`/`render` the 4x reports!

```r
purrr::pmap(.l = list(states, years), .f = render_impact_report)
```

HTML Widgets

Front-end interactivity

```r
library(leaflet)

m <- leaflet() %>%
  addTiles() %>%  # Add default OpenStreetMap map tiles
  addMarkers(lng=174.768, lat=-36.852, popup="The birthplace of R")
m  # Print the map
```

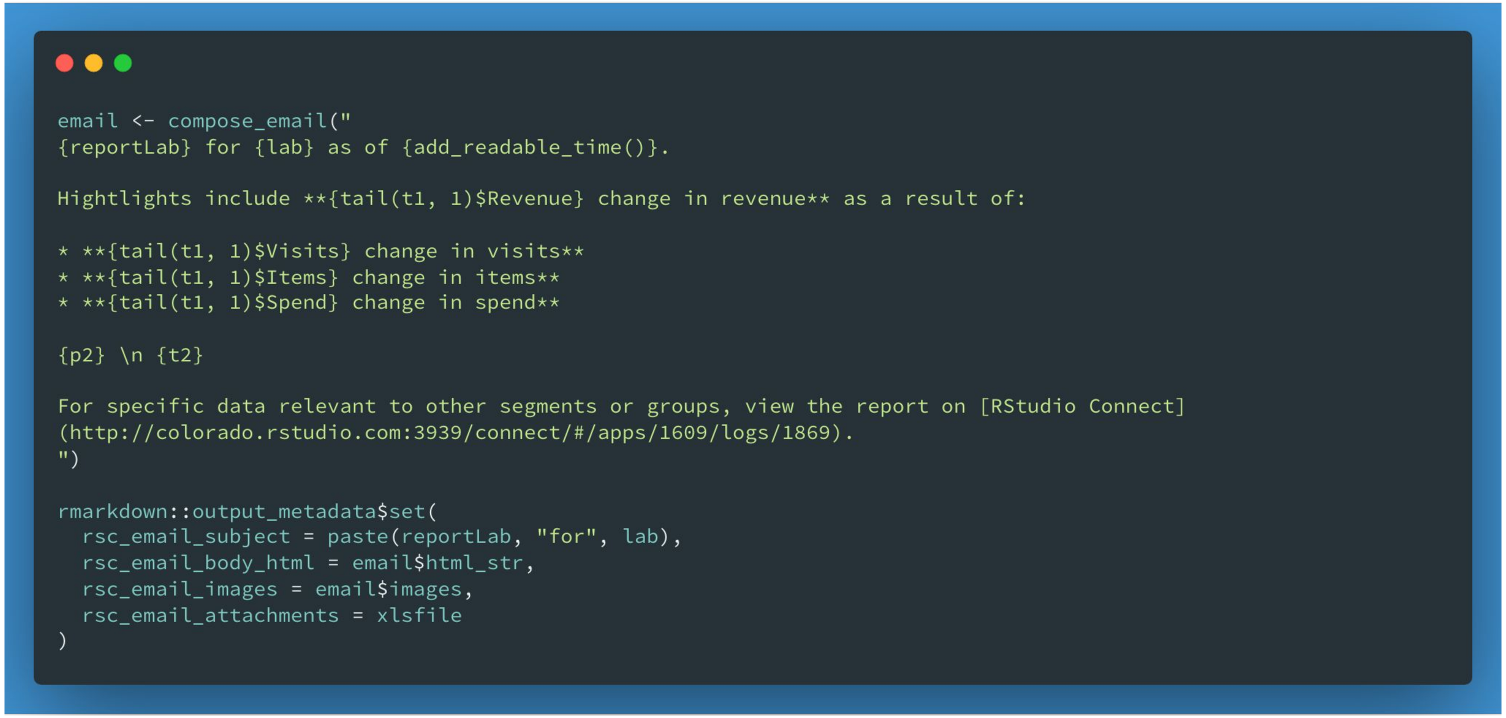


https://www.htmlwidgets.org/

https://rstudio.github.io/crosstalk/

http://gallery.htmlwidgets.org/

Blastula

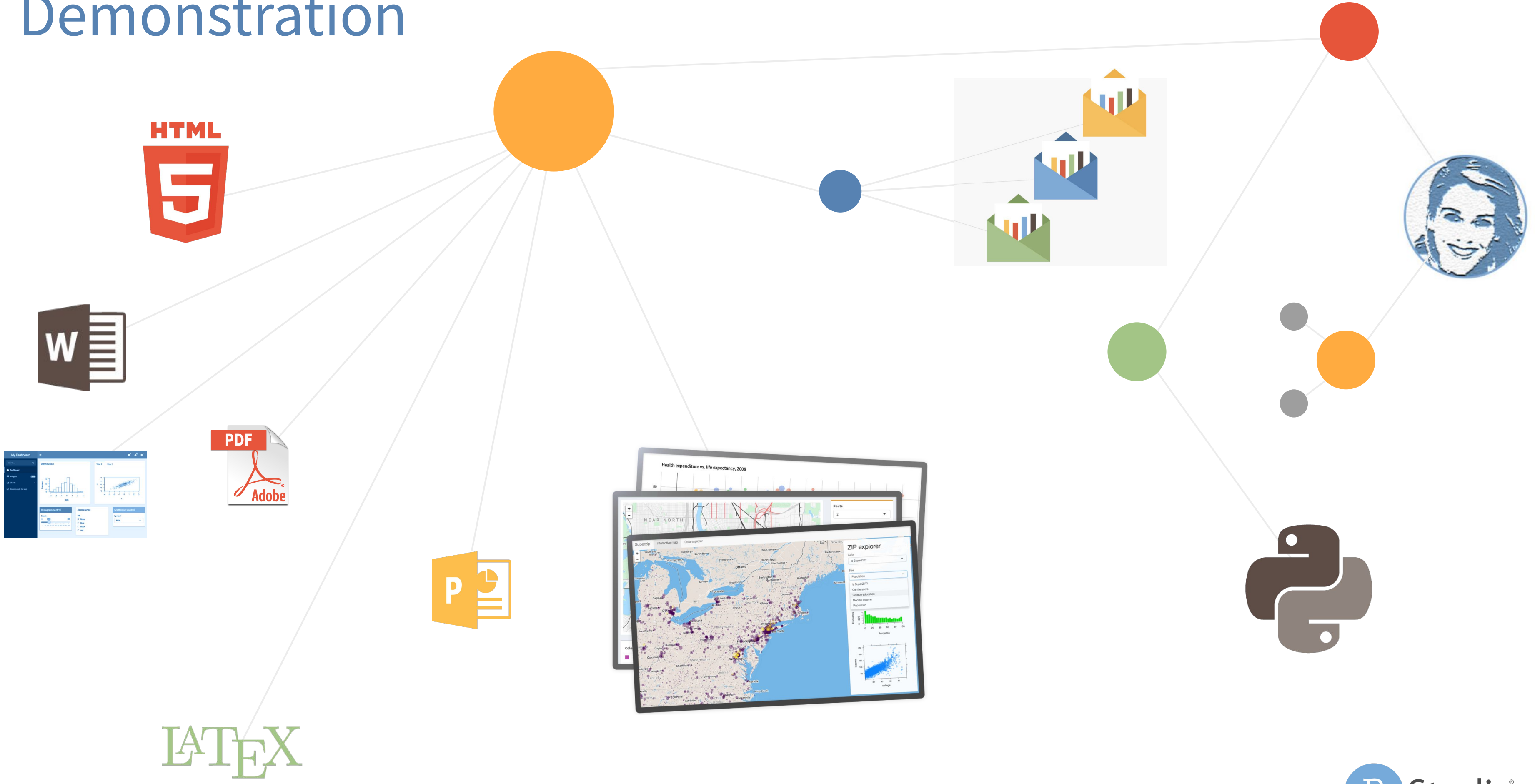


```
email <- compose_email("
{reportLab} for {lab} as of {add_readable_time()}.

Hightlights include **{tail(t1, 1)$Revenue} change in revenue** as a result of:

* **{tail(t1, 1)$Visits} change in visits**
* **{tail(t1, 1)$Items} change in items**
* **{tail(t1, 1)$Spend} change in spend**

{p2} \n {t2}

For specific data relevant to other segments or groups, view the report on [RStudio Connect]
(http://colorado.rstudio.com:3939/connect/#/apps/1609/logs/1869).
")

rmarkdown::output_metadata$set(
  rsc_email_subject = paste(reportLab, "for", lab),
  rsc_email_body_html = email$html_str,
  rsc_email_images = email$images,
  rsc_email_attachments = xlsfile
)
```

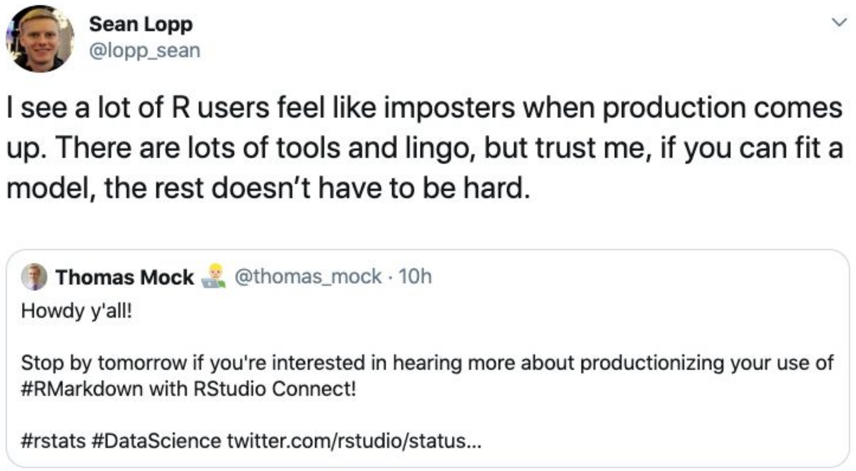
[Custom Scheduled Emails with Blastula]()

RStudio

Demonstration



Connect can help you get started or level up



Sean Lopp
@lopp_sean

I see a lot of R users feel like imposters when production comes up. There are lots of tools and lingo, but trust me, if you can fit a model, the rest doesn't have to be hard.

Thomas Mock 👨‍💻 @thomas_mock · 10h
Howdy y'all!

Stop by tomorrow if you're interested in hearing more about productionizing your use of #RMarkdown with RStudio Connect!

#rstats #DataScience twitter.com/rstudio/status...

Additional topics worth exploring

- Best Practices for Administering R Studio in Production
- Model Management with RStudio Connect
- Shiny in Production: Principles, practices, and tools
- Rstudio Team Quickstart (Try Connect without installing anything)!
- solutions.rstudio.com

Downloads

45 Day Evaluation of Pro Products

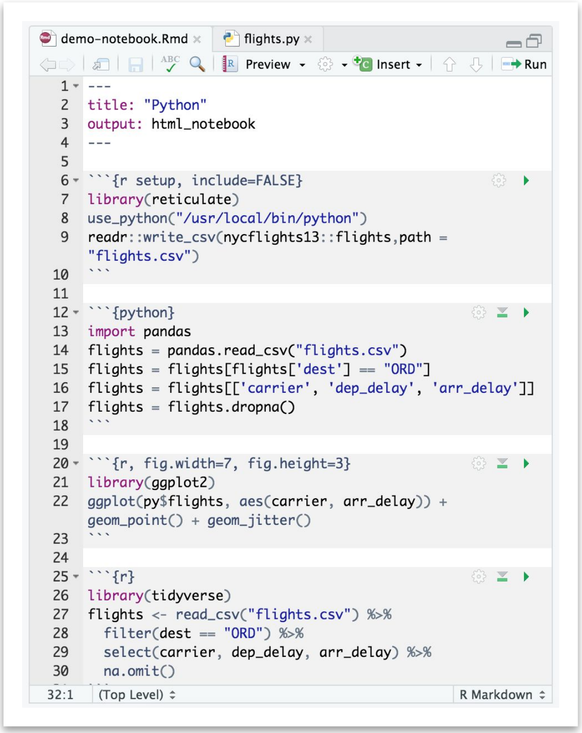
- RStudio Server Pro: https://www.rstudio.com/products/rstudio-server-pro/evaluation/
- RStudio Connect: https://www.rstudio.com/products/connect/evaluation/
- RStudio Drivers: https://www.rstudio.com/products/drivers/drivers-evaluation/

Book a call with us to talk about RStudio Connect:
https://rstudio.youcanbook.me/

Support for multiple languages

Our philosophy is aimed at integrating Python and other languages into R projects

- IDE support for Python, bash, SQL, C++, etc.
- Translate, call, and bind Python from R with the *reticulate* package
- Make Python tools (like TensorFlow and Keras) accessible to R users



~~~
---
title: "Python"
output: html_notebook
---

```{r setup, include=FALSE}
library(reticulate)
use_python("/usr/local/bin/python")
readr::write_csv(nycflights13::flights,path = "flights.csv")
```

```{python}
import pandas
flights = pandas.read_csv("flights.csv")
flights = flights[flights['dest'] == "ORD"]
flights = flights[['carrier', 'dep_delay', 'arr_delay']]
flights = flights.dropna()
```

```{r, fig.width=7, fig.height=3}
library(ggplot2)
ggplot(py$flights, aes(carrier, arr_delay)) + geom_point() + geom_jitter()
```

```{r}
library(tidyverse)
flights <- read_csv("flights.csv") %>%
  filter(dest == "ORD") %>%
  select(carrier, dep_delay, arr_delay) %>%
  na.omit()
~~~

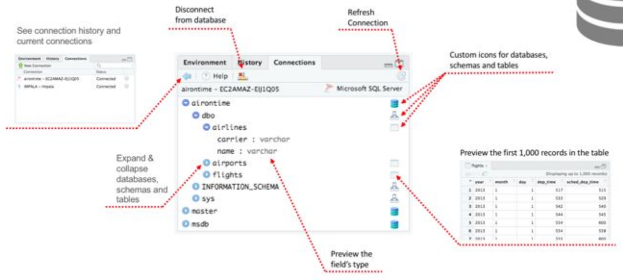
Database with RStudio

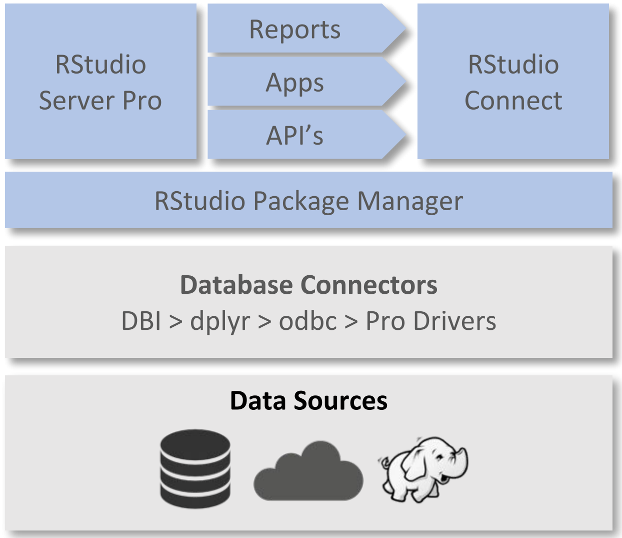
Before improvements to the RStudio toolchain…

- Connections were hard to set up and configure
- Tools and interfaces were inconsistent
- R users had to rely on two tools -- one for data management and one for data analysis
- There was no centralized place to get information

Our solution

- Provide powerful, easy to use tools for administrators
- Offer a new set of open source database connectors
- Improve tooling in the IDE
- Establish a knowledge base for database best practices


See connection history and current connections
Disconnect from database
Refresh Connection
Custom icons for databases, schemas and tables
Environment
History
Connections
Help
airontime - EC2AMAZ-EIJ1Q05
Microsoft SQL Server
airontime
dbo
airlines
carrier : varchar
name : varchar
airports
flights
INFORMATION_SCHEMA
sys
master
msdb
Expand & collapse databases, schemas and tables
Preview the first 1,000 records in the table
Preview the field's type


RStudio Server Pro
Reports
Apps
API's
RStudio Connect
RStudio Package Manager
Database Connectors
DBI > dplyr > odbc > Pro Drivers
Data Sources