

Developing and deploying an R 'app' on AWS

Roman Popat - Data Scientist roman.popat@thedatalab.com

twitter: @rmnppt web: rmnppt.com

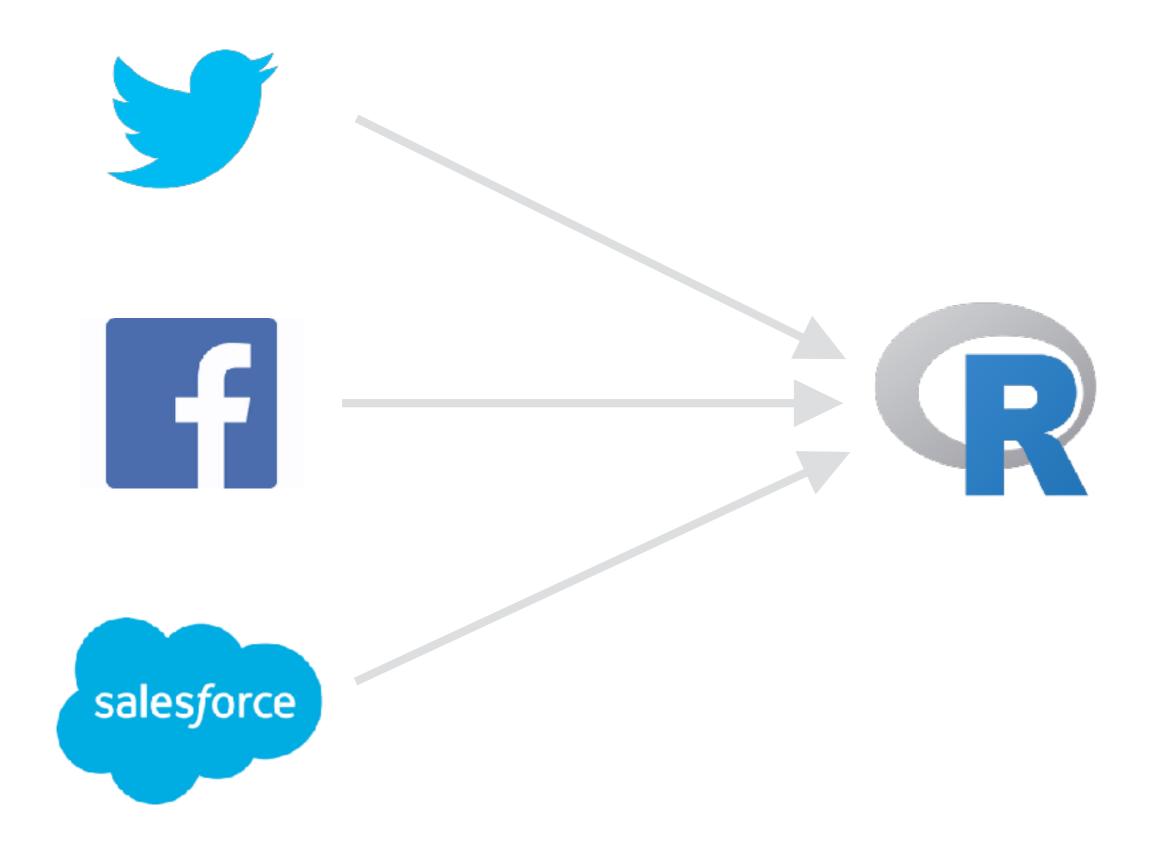
Github: rmnppt



- 1. Accessing an REST API
- 2. Publishing an R package on github
- 3. EC2 instance and R/Rstudio
- 4. RDS storage (MySQL and R)
- 5. Flexdashboard (Rmarkdown)
- 6. Scheduling linux tasks (crontab)

REST API examples





A case study





Login / Register



What does an API call look like?



api.adzuna.com:80/v1/api/jobs/gb/search/
1?

app_id=134859f2&app_key=1d215eccbc34d374
30ce8693d575c51b&what=data%2Bscience





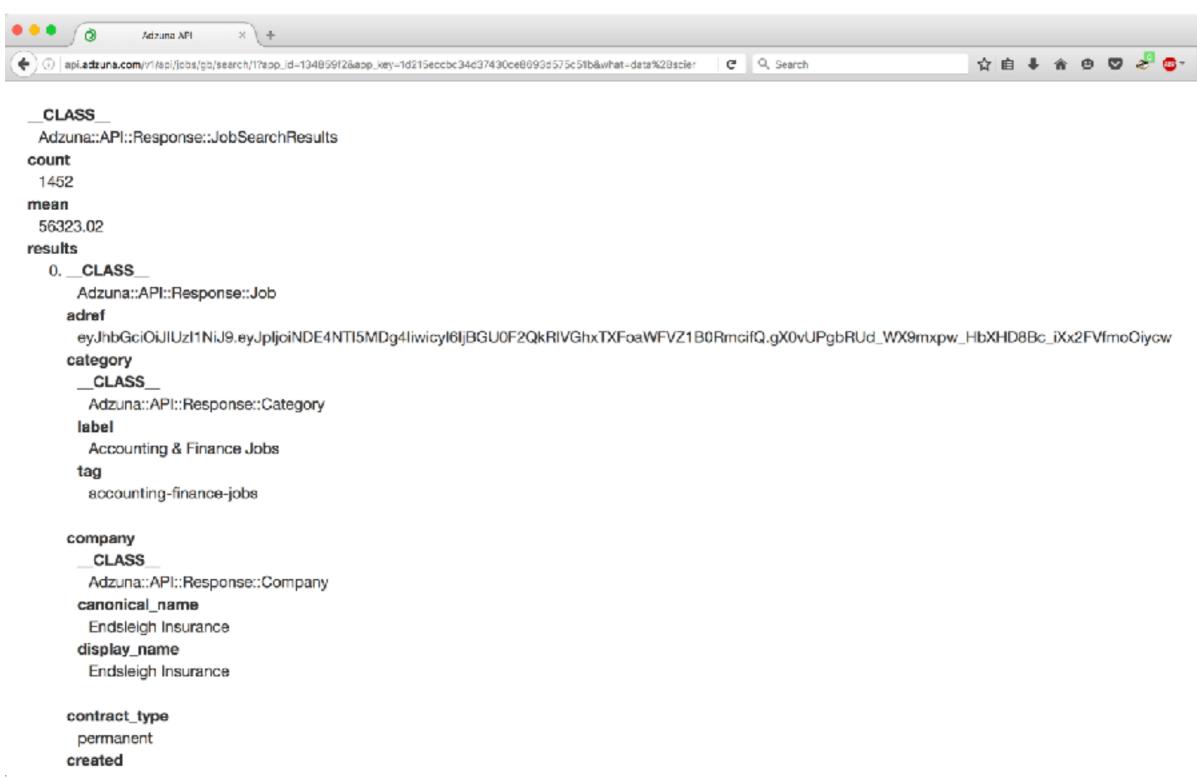
Data returned from API





JSON





The R function

}



```
get_country_page <- function(</pre>
  keyword,
  country,
  app_id,
  app_key,
  page
  this_url <- paste0(
    "http://api.adzuna.com:80/v1/api/jobs/",
    country,
    "/search/",
    page, "?",
    "app_id=", app_id,
    "&app_key=", app_key,
    "&results_per_page=50",
    "&what=", keyword
  dat <- fromJSON(this_url)</pre>
  return(dat)
```



Using the function



```
get_country_page(
   keyword = "data science",
   country = "gb",
   app_id = id,
   app_key = key,
   page = 1
)
```





Sharing Code



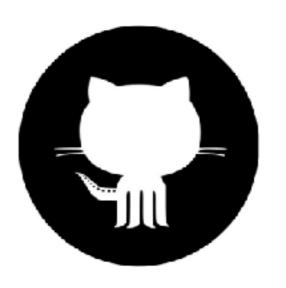
R session

create("adzunar")



Terminal

git init git add git commit git push



Sharing Code



devtools::install_github("rmnppt/adzunar")

library(adzunar)

get_country_page()







- 1. Accessing an REST API
- 2. Publishing an R package on github
- 3. EC2 instance and R/Rstudio
- 4. RDS storage (MySQL and R)
- 5. Flexdashboard (Rmarkdown)
- 6. Scheduling linux tasks (crontab)

R/Rstudio on AWS - EC2







AWS Big Data Blog

Running R on AWS

by Markus Schmidberger | on 23 JUL 2015 | Permalink | Comments

Markus Schmidberger is a Senior Big Data Consultant for AWS Professional Services

Many AWS customers already use the popular open-source statistic software R for big data analytics and data science; others have asked for instructions and best practices for running R on AWS. Several months ago, I wrote a blog post showing you how to connect R with Amazon EMR, install RStudio on the Hadoop master node, and use R packages such as *rmr2* or *plyrmr* to analyze a huge public weather data set. In this post, I show you how to install and run R, RStudio Server, and Shiny Server on AWS.

RStudio is a popular IDE, licensed either commercially or under AGPLv3, for working with R. This is ideal if you don't want to connect to a server via SSH and use code editors such as vi to do analytics. RStudio is available in a desktop version, or a server version that allows you to access R via a web browser.

After you've analyzed your results, you may want to visualize them. Shiny is a great R package, licensed either commercially or under AGPLv3, that you can use to create interactive dashboards. Shiny provides a web application framework for R. It turns your analyses into interactive web applications; no HTML, CSS, or JavaScript knowledge required. Shiny Server can deliver your R visualization to your customers via a web browser and execute R functions, including database queries, in the background.

The examples in this post use the AWS public data set CCAFS-Climate Data, a 6 TB data set with high-resolution climate data, to assess the impacts of climate change, primarily on agriculture. The image below shows what the architecture will look like.

https://aws.amazon.com/blogs/big-data/running-r-on-aws/



- 1. Accessing an REST API
- 2. Publishing an R package on github
- 3. EC2 instance and R/Rstudio
- 4. RDS storage (MySQL and R)
- 5. Flexdashboard (Rmarkdown)
- 6. Scheduling linux tasks (crontab)



- 1. Accessing an REST API
- 2. Publishing an R package on github
- 3. EC2 instance and R/Rstudio
- 4. RDS storage (MySQL and R)
- 5. Flexdashboard (Rmarkdown)
- 6. Scheduling linux tasks (crontab)

OAA - Obscure Amazon Acronyms



AWS - Amazon Web Services

EC2 - Elastic Cloud Compute

S3 - Simple Storage Service

RDS - Relational Database Service

finished



