

AWS and R

Twin Cities R User Group (TCRUG)
Sept. 20, 2018

LeSean Bruneau

Twitter: @leseanbruneau

LinkedIn: <http://www.linkedin.com/in/lesean-bruneau-7092625>

Github: <https://github.com/mndatascienceexamples/AWS-and-R>

Blog: <http://datascienceexamples.com>

Agenda

- AWS Services for demo projects
- Overview demo projects
- Project 1 – RStudio Local and AWS S3
 - RStudio local integration with AWS S3
 - Display data results on AWS S3 static web site
- Project 2 – RStudio Server on AWS EC2
 - Install R on AWS EC2 instance
 - Install RStudio Server on AWS EC2 instance
 - Run RStudio Server on EC2 Instance

AWS Services

- EC2 Instance (Compute)
 - Linux server
 - Security group permissions
- IAM (Authentication and Authorization)
- S3 Bucket (Storage)
 - Directories and Files
 - Permissions
 - Assign permissions on files/directories
 - Use IAM for program access to S3 Bucket

Project 1 Demo Overview

- All MLB 2017 Regular Season Games
 - <http://baseball-reference.com>
- S3 Bucket for data input file
- RStudio Local
 - Create dataframe from data input file in S3
 - R Function to select one team's games
 - Output – write JSON file to S3 Bucket
- S3 Bucket for static web application
 - Display output from RStudio desktop on web page

Project 2 Demo Overview

- Create AWS EC2 Instance
- Install R and R system packages
- Install RStudio Server
- Run RStudio Server on a web browser

Project Setup Information - Local

- Create local workspace
 - Demo: OS: Windows; Directory: C:\R directory
- Github Repo
 - <https://github.com/mndatascienceexamples/AWS-and-R>
 - Extract zip file to local workspace (C:\R\AWS-and-R-master)

AWS S3 Information

- Object Storage, not Block Storage
- AWS S3 Objects
 - Key Name (file name – unique id within a bucket)
 - file data
 - (optional) file version
 - `http://<<S3_BUCKET_NAME>>.s3.amazonaws.com/<<KEY_NAME>>`
- REST APIs for AWS S3 Operations
 - Read / Write / Delete an object
 - List keys in a bucket
- Static Web Site using HTTP

Project Setup Information - AWS

- **AWS S3 Bucket**
 - Create S3 Bucket
 - Upload Github R directory (c:\R\AWS-and-R-master\R)
 - AWS S3 upload default options
 - Upload Github webapp directory (c:\R\AWS-and-R-master\webapp)
 - Public Read-only directory access
 - All other AWS S3 upload default options
- **AWS IAM User Account**
 - Access Type: Programmatic access
 - Policy Name: PowerUserAccess
 - Save Secret Key and Access Key (download credentials.csv file)

Project 1 – RStudio and S3 Setup

- RStudio Desktop

- Set working directory

```
R> setwd("C:\\R")
```

- Set System Env Variables for AWS IAM Account

```
<<Template>>
```

```
R> Sys.setenv("AWS_ACCESS_KEY_ID" = "<PUT-ACCESS-KEY>", "AWS_SECRET_ACCESS_KEY" = "<PUT-SECRET-KEY>")
```

```
<<Sample>>
```

```
R> Sys.setenv("AWS_ACCESS_KEY_ID" =  
"DRFAIY3EKDHLZW5YMMXA", "AWS_SECRET_ACCESS_KEY" =  
"AMeLhWHUw12S9wHUI+4XyhGGaSiluVOftSk8kUvC")
```

- Install R Packages (aws.s3)

```
R> source("AWS-and-R-master\\R\\local\\Install_Libraries.R")
```

Project 1 – RStudio and S3

- Load R Libraries

```
R> library("aws.s3")  
R> library("jsonlite")
```

- Create function – select team's games

```
R> teamGames <- function(games, team) { ... }
```

- Load data

```
R> mlb_2017 <- aws.s3::s3read_using  
(read.csv, object = "s3://<YOUR_S3_BUCKET_NAME>/R/  
2017-mlb-games.txt", sep=",", header=FALSE, stringsAsFactors =  
TRUE, quote="\\"", comment.char = "")
```

- Add header information

```
R> names(mlb_2017) <- c("GmNo", ... , "Orig. Scheduled")
```

Project 1 – RStudio and S3

- Execute R Function with Team Name Abbr.

```
R> tg <- teamGames(mlb_2017,"MIN")
```

- Write results to JSON file

```
R> write_json(tg,"c:\\R\\data.json", pretty = TRUE)
```

- Upload JSON file to S3

```
R> put_object(file = "c:\\R\\data.json", object =  
"webapp/js/data.json", bucket = "<YOUR_S3_BUCKET_NAME>",  
acl = c("public-read"))
```

- Verify results in web application

Web Browser URL

https://s3.amazonaws.com/<YOUR_S3_BUCKET_NAME>/webapp/index.html

Project 2 – RStudio Server EC2 Setup

- Create EC2 Instance

- AMI: Amazon Linux 2 AMI (HVM), SSD Volume Type
- Instance Type: General Purpose t2.micro
- Configuration Instance: <<default options>>
- Add storage: <<default options>>
- Add tags: {Key: Name; Value: R Compute Server}
- Configure Security Group – Create group opening following ports
 - SSH – Port Range: Port 22 – Source: <YOUR_IP_ADDRESS>/32
 - Custom TCP Rule – Port Range: Port 8787 – Source: 0.0.0.0/0
- Review: <<default options>>; Launch Instance
- Key Pair: New or existing

Project 2 – RStudio Server EC2 Setup

- EC2 Instance Update
 - > **sudo yum -y update**
- EC2 Instance – Install Git
 - > **sudo yum -y install git**
- EC2 Instance – Clone Git Repo
 - From ec2-user home directory (/home/ec2-user)
 - > **git clone <https://github.com/mndatascienceexamples/AWS-and-R>**

Project 2 – RStudio Server Install

Four Scripts for R, RStudio Server installation

(/home/ec2-user/AWS-and-R/R/server)

- **Script1_install_ec2_utils_sudo.sh**
 - Install Linux Operating System dependencies for R installation
- **Script2_download_R_utils.sh**
 - Download R installation package and configure installation
- **Script3_install_R_sudo.sh**
 - R installation on Linux server
- **Script4_install_R_packages_sudo.sh**
 - R system libraries installation on Linux server

Project 2 – RStudio Server Install

Script 1 – Linux OS Dependencies

```
~ > sudo /home/ec2-user/AWS-and-R/R/serverScript1_install_ec2_utils_sudo.sh
```

```
yum -y install make libX11-devel.* libICE-devel.* libSM-devel.* libdmx-devel.* libx*
```

```
yum -y install xorg-x11* libFS* libX* readline-devel gcc-gfortran gcc-c++
```

```
yum -y install texinfo tetex texlive texlive-latexyum -y install bzip2-devel.x86_64 bzip2-libs.x86_64  
bzip2.x86_64
```

```
yum -y install libcurl libcurl-develyum -y install pcre pcre-devel
```

```
yum -y install java-1.8.0-openjdk.x86_64 java-1.8.0-openjdk-devel.x86_64yum -y install openssl-  
devel
```

Project 2 – RStudio Server Install

Script 2 – Download R Installation Packages

```
~ > /home/ec2-user/AWS-and-R/R/server/Script2_download_R_utils.sh
```

```
mkdir ~/lib
```

```
cd ~/lib
```

```
wget http://cran.r-project.org/src/base/R-3/R-3.5.1.tar.gz
```

```
wget https://download2.rstudio.org/rstudio-server-rhel-1.1.456-x86\_64.rpm
```

```
...
```

```
tar xzf R-3.5.1.tar.gz && cd R-3.5.1
```

```
./configure --prefix=/opt/R/3.5.1/bin/R --with-x=no --enable-R-shlib
```

```
make
```

```
make pdf
```

```
make info
```


Project 2 – RStudio Server Install

Script 3 – R Installation on Linux Server

```
~ > sudo /home/ec2-user/AWS-and-R/R/server/Script3_install_R_sudo.sh
```

```
cd /home/ec2-user/lib/R-3.5.1
```

```
make install
```

```
ln -s /opt/R/3.5.1/bin/R/bin/R /usr/bin/R
```

```
ln -s /opt/R/3.5.1/bin/R/bin/R /usr/local/bin/R
```

```
export PATH=$PATH:/opt/R/3.5.1/bin/R/bin
```

```
export RSTUDIO_WHICH_R=/opt/R/3.5.1/bin/R/bin/R
```

```
cd /home/ec2-user/lib
```

```
yum -y install --nogpgcheck rstudio-server-rhel-1.1.456-x86_64.rpm
```

```
rstudio-server verify-installation
```

Project 2 – RStudio Server Install

Script 4 – R System Libraries Installation

```
~ > sudo /home/ec2-user/AWS-and-R/R/server/Script4_install_R_packages_sudo.sh
```

```
R CMD INSTALL /home/ec2-user/lib/jsonlite_1.5.tar.gz
```

```
R CMD INSTALL /home/ec2-user/lib/mime_0.5.tar.gz
```

```
R CMD INSTALL /home/ec2-user/lib/curl_3.2.tar.gz
```

```
R CMD INSTALL /home/ec2-user/lib/openssl_1.0.2.tar.gz
```

```
R CMD INSTALL /home/ec2-user/lib/R6_2.2.2.tar.gz
```

```
...
```

```
R CMD INSTALL /home/ec2-user/lib/aws.signature_0.4.4.tar.gz
```

```
R CMD INSTALL /home/ec2-user/lib/aws.s3_0.3.12.tar.gz
```

Project 2 – RStudio Server User

Create user on Linux server for RStudio Server

> sudo adduser rstudio

> sudo sh -c “echo rstudio | passwd rstudio --stdin”

Project 2 – RStudio Server Connect

Web Browser – Connect to RStudio Server

- `http://<EC2_INSTANCE_IP_ADDRESS>:8787`
- Login with Linux user `rstudio` and password
- Check RStudio Server working directory