

# E-rate Funding: How I Built a Data Pipeline in R

## From a Scheduled R Script on AWS to a Shiny Dashboard

---

Bree Norlander, Data Scientist  
[norlab@uw.edu](mailto:norlab@uw.edu)

Chris Jowaisas, Senior Research Scientist  
[chrisjow@uw.edu](mailto:chrisjow@uw.edu)

# FCC Schools and Libraries Program aka E-rate

## E-rate

Ensuring that schools and libraries across the U.S. are connected to information and resources through the internet



Photo from the USAC E-rate website: <https://www.usac.org/e-rate/>

## The Problem

---

- > Pulling the data from the API can take > 30 minutes
- > Datasets are updated daily
- > Datasets are large & require transformations before visualizing in a Shiny App

## The Solution

---

- > **A virtual machine on AWS**
- > **A cron job that runs the R Script daily**
- > **AWS EC2 Instance on a Schedule**
- > **Transformed datasets stored in AWS S3**
- > **Shiny App that pulls directly from S3 storage**



Search in this product

Eng

AWS > Documentation > Amazon EC2

# Amazon Elastic Compute Cloud Documentation

Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides resizable computing capacity—literally, servers in Amazon's data centers—that you use to build and host your software systems.

## Amazon EC2

### User Guide for Linux Instances

Describes key concepts of Amazon EC2 and provides instructions for using the features of Amazon EC2.

[HTML](#) | [PDF](#) | [Kindle](#) | [GitHub](#)

### User Guide for Windows Instances

Describes key concepts for Amazon EC2 and provides instructions for launching and using your Windows instance.

[HTML](#) | [PDF](#) | [Kindle](#) | [GitHub](#)

### User Guide for AWS Nitro Enclaves

Describes key concepts for AWS Nitro Enclaves and provides instructions for using enclaves.

[HTML](#) | [PDF](#)

### API Reference

Documents the Amazon EC2 Query API.

[HTML](#) | [PDF](#)

### EC2 section of the AWS CLI Reference

Documents the AWS CLI commands for Amazon EC2.

[HTML](#)

### EC2 Instance Connect API Reference

Documents the Amazon EC2 Instance Connect Query API.

[HTML](#) | [PDF](#)

## Helpful Documentation:

- [https://docs.aws.amazon.com/ec2/index.html?nc2=h\\_ql\\_doc\\_ec2](https://docs.aws.amazon.com/ec2/index.html?nc2=h_ql_doc_ec2)
- [https://www.louisaslett.com/RStudio\\_AMI/](https://www.louisaslett.com/RStudio_AMI/)

```
1 print(Sys.time())  
2  
3 # Load libraries  
4 library(RSocrata)  
5 library(dplyr)  
6 library(tidyr)  
7 library(ggplot2)  
8 library(stringr)  
9 library(glue)  
10 library(purrr)  
11 library(fuzzyjoin)  
12 library(geosphere)  
13 library(janitor)  
14 library(stringdist)  
15 library(aws.s3)  
16 library(aws.ec2metadata)  
17  
18 # Pull disbursement data for all funded entities  
19 disbursement <- read.socrata(  
20   "https://opendata.usac.org/resource/qdmp-ygft.json?form_471_frn_status_name=Funded",  
21   app_token = Sys.getenv("USAC_Socrata")  
22 )  
23  
24 # Write to s3 bucket  
25 s3write_using(disbursement, FUN = write.csv, bucket = "erate-data/data", object = "Funded_Disbursements.csv")  
26  
27 print("disbursement dataset written to s3")  
28  
29 print("disbursement dataset written to s3")  
30
```

```
# Edit this file to introduce tasks to be run by cron.  
#  
# Each task to run has to be defined through a single line  
# indicating with different fields when the task will be run  
# and what command to run for the task  
#  
# To define the time you can provide concrete values for  
# minute (m), hour (h), day of month (dom), month (mon),  
# and day of week (dow) or use '*' in these fields (for 'any').#  
# Notice that tasks will be started based on the cron's system  
# daemon's notion of time and timezones.  
#  
# Output of the crontab jobs (including errors) is sent through  
# email to the user the crontab file belongs to (unless redirected).  
#  
# For example, you can run a backup of all your user accounts  
# at 5 a.m every week with:  
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/  
#  
# For more information see the manual pages of crontab(5) and cron(8)  
#  
# m h  dom mon dow   command  
#  
PATH="/usr/bin:/bin"  
HOME="/home/ubuntu"  
SHELL="/bin/sh"  
10 12 * * MON-FRI Rscript erate_data_script.R > temp.log 2>&1 | aws s3 cp temp.log s3://erate-data/logs/  
15 13 * * MON-FRI aws s3 cp temp.log s3://erate-data/logs/
```

- Mike Page Tutorial for help with crontab (bottom of page):  
<https://www.mikejohnpage.com/blog/returning-tweets-on-a-schedule-in-r-using-aws-ec2-rds-and-cron/>
- It took many tries to get the crontab to correctly use the AWS CLI. I ended up dumping the env and then adding those variables to the crontab. This website helped me get there:  
<https://krisjordan.com/blog/2013/11/04/timesaving-crontab-tips>

^G Get Help   ^O Write Out   ^W Where Is   ^K Cut Text   ^J Justify   ^C Cur Pos   M-U Undo  
^X Exit   ^R Read File   ^\ Replace   ^U Uncut Text   ^T To Spell   ^\_ Go To Line   M-E Redo

v X Exit   v B Read   v E Edit   v / a Replace   v S Insert   v U Undo   v T Text   v C Cut Text   v U Paste   v G Go To Line   v M Undo  
v C Get Help   v D Delete   v F Find   v I Insert   v W Write   v S Insert   v M Undo   v U Paste   v G Go To Line   v M Undo

## CloudWatch Events is now Amazon EventBridge

Amazon EventBridge (formerly CloudWatch Events) provides all functionality from CloudWatch Events and also launched new features such as Custom event buses, 3rd party event sources and Schema registry to better support our customers in the space of event-driven architecture and applications.

[Amazon EventBridge documentation](#)

## Rules

Rules route events from your AWS resources for processing by selected targets. You can create, edit, and delete rules.

[Create rule](#)

Actions ▾



Status	All	Name	Description
<input type="radio"/>		StartErateEC2Instance	start EC2 instance for eRate at 1200 UTC every weekday
<input type="radio"/>		StopErateEC2Instance	stop EC2 instance for eRate at 1400 UTC every weekday

Schedule Cron expression `0 12 ? * MON-FRI *`

Next 10 Trigger  
Date(s)

1. Mon, 02 Nov 2020 12:00:00 GMT
2. Tue, 03 Nov 2020 12:00:00 GMT
3. Wed, 04 Nov 2020 12:00:00 GMT
4. Thu, 05 Nov 2020 12:00:00 GMT
5. Fri, 06 Nov 2020 12:00:00 GMT
6. Mon, 09 Nov 2020 12:00:00 GMT
7. Tue, 10 Nov 2020 12:00:00 GMT
8. Wed, 11 Nov 2020 12:00:00 GMT
9. Thu, 12 Nov 2020 12:00:00 GMT
10. Fri, 13 Nov 2020 12:00:00 GMT

Status Enabled

Description start EC2 instance for eRate at 1200 UTC every weekday

- <https://aws.amazon.com/premiumsupport/knowledge-center/start-stop-lambda-cloudwatch/>
- <https://docs.aws.amazon.com/solutions/latest/instance-scheduler/welcome.html>

## Merging E-Rate Data and IMLS PLS Data

This page is a work-in-progress. We intend to bring you insights from our work merging the E-Rate data with the Institute for Museum and Library Services (IMLS) Public Library Survey (PLS) [data](#). (We are currently using 2018 IMLS PLS data.) There are no perfectly matching columns between the two datasets so we have designed a methodology that uses both geolocation matching and character matching to combine the data. It is not perfect but we think it's a really good start. While you are waiting for new graphs and charts to appear on this page, perhaps you would like to peruse our 'decoder' dataset. Below you can download the matches we have made between E-Rate and IMLS data. Enjoy!

### Download E-Rate Entities and IMLS Matches

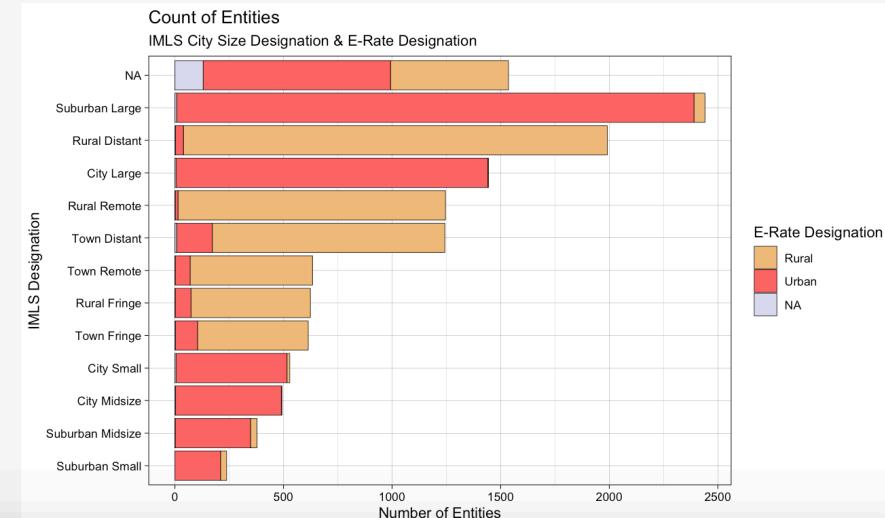
[!\[\]\(147b0c7dce349edf02b6b21226344f99\_img.jpg\) Download \(.csv\)](#)

Show **10**  entries

Entity designations from both IMLS and E-Rate data for entire E-Rate dataset (2016-2020).

**IMLS Designation (LOCALE\_ADD) ◆ E-Rate Designation ◆ Count of Unique Entities**

City Large	Rural	3
City Large	Urban	1436
City Large	NA	5
City Midsize	Rural	3
City Midsize	Urban	487
City Midsize	NA	3
City Small	Rural	13
City Small	Urban	11
City Small	NA	1
Suburban Large	Urban	2400



# E-rate and Libraries Dashboard:

<https://uwtascha.shinyapps.io/eRate-dashboard/>

## Code and Slides:

<https://github.com/tascha/GASP2020>

---

Bree Norlander, Data Scientist  
norlab@uw.edu

Chris Jowaisas, Senior Research Scientist  
chrisjow@uw.edu