How to Run a Large Number of R files using Amazon Web Services on the Cloud

Mark W. Miller

February 21, 2021

This is a tutorial explaining how to run a large number of R files using Amazon Web Services (AWS). The learning curve for AWS is steep and I have found no instructions elsewhere for running a large number of R files with it. This tutorial attempts to make the entire process clear via step-by-step instructions. I am virtually entirely self-taught with AWS. Efficiency of the procedures below surely can be improved, probably greatly. Nevertheless, they do seem to work as is. I can be reached via: RtutorialAWS@gmail.com

I am using a Windows 10 laptop and will use a command line interface to run R on the cloud. If you wish to use a Macintosh computer some of the steps I provide below might be different. I have never used a Macintosh computer with AWS. Nor have I ever attempted to use RStudio with AWS.

I will begin by creating a new AWS account, which is free. Running the example in this tutorial with Spot Instances in AWS might cost $0.05. I will restrict my example to 20 very simple R files. However, the approach illustrated here works when using, for example, 500 R files and likely far more. I also will use two applications called PuTTY and FileZilla both of which are free. In AWS I will use something called spot instances which is the cheapest way to use AWS. When running hundreds of R files you likely will need to pay some money, maybe $5.00 - $20.00 US depending on how long the files take to run.

I have created 20 R files and 20 input data files in CSV format. The R files are divided into two groups of ten. One group is stored in the folder ‘R\_files\_a’ and the other group is stored in the folder ‘R\_files\_b’. Each R file in the folder ‘R\_files\_a’ will read a unique input data file in the folder ‘Input\_data\_files\_a’. Similarly, each R file in the folder ‘R\_files\_b’ will read a unique input data file in the folder ‘Input\_data\_files\_b’. These input data files are in CSV format. Each R file simply reads two columns of numbers in an input data file, adds those numbers by row, and outputs the result to a different data file also in CSV format. The R files in the folder ‘R\_files\_a’ will output results to the folder ‘Output\_data\_files\_a’; the R files in the folder ‘R\_files\_b’ will output results to the folder ‘Output\_data\_files\_b’.

I have kept the operations in R simple so I can check quickly whether the output is correct. However, I use parallel processing to run the R files. The R file ‘master\_R\_file\_a.r’ implements parallel processing for the folder ‘R\_files\_a’; the R file ‘master\_R\_file\_b.r’ implements parallel processing for the folder ‘R\_files\_b’. The bash file ‘bash\_file\_a.sh’ submits ‘master\_R\_file\_a.r’; the bash file ‘bash\_file\_b.sh’ submits ‘master\_R\_file\_b.r’.

I have now created all of the files and folders needed to run all 20 R files. All of the folders and files mentioned above are included in this tutorial. Even the output files are included. Before working through the tutorial I recommend deleting these output files or moving them to different folders. If you lose these output files you can always download them again from my GitHub site.

The next step is to create an AWS account. To do so:

1. Go to: <https://aws.amazon.com/>
2. Click the orange tab ‘Sign in to the Console’ in the upper-right corner of the screen
3. Click ‘Create a new AWS account’
4. Provide:
   1. An email address
   2. A password
   3. A user name
   4. Click ‘Continue (step 1 of 5)’
5. Provide contact information
   1. Click ‘Continue (step 2 of 5)’
6. Provide billing information
   1. Click ‘Verify and Continue (step 3 of 5)’
   2. This example will not require paying any money
7. Confirm your identity
   1. Click ‘Send SMS (step 4 of 5)’
   2. Provide the code you received
   3. Click ‘Continue (step 4 of 5)’
8. Click ‘Complete sign up’
   1. I left the default ‘Basic support – Free’ selected.
9. Click ‘Go to the AWS Management Console’
10. Sign into the newly-created AWS account
    1. Enter Root user email address
       1. I used the same email address here that I used above
    2. Click ‘Next’
    3. Enter a password
    4. Click ‘Sign in’
11. Basic features of this account can be used free for one year. However, Spot Instances might cost money even when using instance types that are free otherwise.

The next steps are to create a key pair and a security group in AWS. Then install PuTTY and configure the key pair for use with PuTTY. I will cover these steps in the outline below except that PuTTY is already installed on my laptop. If my steps are incomplete or confusing you can watch video of how to do these steps in the first three sections of Stephane Maarek’s Udemy course “Amazon EC2 Master Class (with Auto Scaling & Load Balancer)”. Those videos will show you how to create an AWS account (which I have already done above) and how to use PuTTY with your AWS account. Those videos will also cover topics like key pairs and security groups. You can buy Stephane Maarek’s course probably for under $15.00 when it is on sale, which it frequently is. I do not recommend paying the full price, which is over $100.00, unless my instructions below are inadequate and you are in a huge hurry. Watching the first three sections of his course might take 1.5 hours. Full disclosure, I have not finished watching Section 3 myself. His course does not cover using R as far as I know. If you want to buy his course go the Udemy website: <https://www.udemy.com/> and search for the course name provided above. If you have already purchased his course here is a link to his first video: <https://www.udemy.com/course/aws-ec2-masterclass/learn/lecture/9197190#overview> Note that Stephane Maarek mentions much more in his introductory videos than I cover here. Do not be alarmed or intimidated if you watch his videos, which are quite good. I provide the minimum information needed to run this R example.

Here are the steps to use to create a key pair and a security group. We will be using Amazon EC2 which stands for Elastic Cloud Computing. I launch a free instance to create a key pair and security group then terminate that instance and use the key pair and security group next with my tutorial example.

1. I am in Region US East (Ohio) us-east-2 by default which I can see by clicking on the triangle next to the word Ohio in the upper right next to my name.
   1. I suspect my example will work in any region in the US, Canada or Europe. You can chose any region you wish.
2. Click on ‘Services’ in the upper-left corner of the screen
   1. Click on EC2 which is the first option in the Compute column
3. Scroll down a little and click the large orange ‘Launch instance’ button
   1. Click ‘Launch instance’ again this time under the orange button
4. Historically I always have selected the following AMI which requires scrolling down quite a bit on the ‘Step 1: Choose an Amazon Machine Image (AMI)’ screen
   1. Ubuntu Server 18.04 LTS (HVM), SSD Volume Type
   2. Click the blue ‘Select’ button to the right of the AMI name
5. The free-tier instance type ‘t2.micro’ is selected by default on the ‘Step 2: Choose an Instance Type’ screen
   1. I will use this instance type to create the key pair and security group
   2. Click ‘Next: Configure Instance Details’
6. On the ‘Step 3: Configure Instance Details’ screen keep everything at their default settings
   1. Click ‘Next: Add Storage’
7. Make no changes on the ‘Step 4: Add Storage’ screen
   1. Click ‘Next: Add Tags’
8. Make no changes on the ‘Step 5: Add Tags’ screen
   1. Click ‘Next: Configure Security Group’
9. Creating a security group is important
   1. On the ‘Step 6: Configure Security Group’ screen the default setting is ‘Create a new security group’
   2. For ‘Security group name:’ I will use ‘Rtutorial’ but you can use any name you want
   3. Keep the remaining defaults. Importantly, I want the following default values for interaction with PuTTY later:
      1. Type SSH
      2. Port Range 22
   4. Click ‘Review and Launch’
10. On the ‘Step 7: Review Instance Launch’ screen click the ‘Launch’ button
11. I am asked to select an existing key pair or create a new key pair
    1. I select ‘Create a new key pair’ with this name: Rtutorial
    2. Click ‘Download Key Pair’
    3. The full key pair name is: Rtutorial.pem
    4. I put this key pair in the folder containing all of the earlier folders and files that I created
    5. This key pair is important and you must keep it secure. Back it up somewhere and do not share it with anyone.
    6. Click ‘Launch Instances’
12. On the ‘Launch Status’ screen scroll down
    1. Click ‘View Instances’
13. The instance is ‘running’ as shown under ‘Instance state’

The next step we need to complete is installation of PuTTY and conversion of our key pair to a format PuTTY can use. PuTTY is an application through which we give commands from our existing physical computer to our virtual instance on the cloud.

1. To download PuTTY go to: <https://www.putty.org/>
2. Click on the word ‘here’ in this line: ‘You can download PuTTY here.’
3. The 32-bit version of PuTTY supposedly works with ‘all processors and on all versions of Windows that PuTTY supports’. That file is: putty-0.74-installer.msi
   1. However, apparently I used the 64-bit version of PuTTY on my laptop:
      1. putty-64bit-0.74-installer.msi
   2. Select whichever version of PuTTY you think is appropriate. There is a FAQ page on the PuTTY website to help you decide.
   3. <https://www.chiark.greenend.org.uk/~sgtatham/putty/faq.html#faq-32bit-64bit>
4. I do not wish to reinstall PuTTY on my laptop.
   1. The section of Stephane Maarek’s Udemy course on setting up PuTTY is called ‘How to SSH into EC2 using Windows – Hands On’
   2. According to that video installing PuTTY is extremely straight-forward like installing any typical Windows application.
5. The key pair I downloaded is called ‘Rtutorial.pem’. I need to convert that file to a different format called ‘Rtutorial.ppk’
   1. Stephane Maarek’s Udemy course section ‘How to SSH into EC2 using Windows – Hands On’ describes this procedure in detail.
   2. You can also find instructions here:
      1. <https://aws.amazon.com/premiumsupport/knowledge-center/convert-pem-file-into-ppk/>
   3. I describe the procedure next
6. You only need to convert a pem key to a ppk key once.
   1. Open PuTTYgen
      1. This program was installed without you knowing it when you installed PuTTY
   2. Click ‘Load’ next to ‘Load an existing private key file’
   3. Load the file ‘Rtutorial.pem’
   4. Click ‘OK’
   5. Click ‘Save private key’
   6. Click ‘Yes’
   7. I used the name ‘Rtutorial.ppk’
   8. Click ‘Save’
   9. Close PuTTYgen

Our next step is to connect PuTTY to our instance using the key pair we just created.

1. Open PuTTY
2. Click the square to the left of the Instance ID of the running instance in AWS
   1. The square will turn blue
3. In AWS scroll down and copy the ‘Public IPv4 DNS’ of the running instance
   1. ec2-1-23-456-789.us-east-2.compute.amazonaws.com
4. In PuTTY in the ‘Host Name (or IP address)’ box paste the following
   1. [ubuntu@ec2-1-23-456-789.us-east-2.compute.amazonaws.com](mailto:ubuntu@ec2-1-23-456-789.us-east-2.compute.amazonaws.com)
   2. This is the Public IPv4 DNS preceded by ‘ubuntu@’
5. Click the word ‘Connection’ in PuTTY in the left-hand column until you see a box labeled
   1. Seconds between keepalives (0 to turn off)
   2. I put 180 in there
6. Click ‘SSH’ in the left-hand column in PuTTY
   1. Click ‘Auth’
   2. In the box ‘Private key file for authentication:’ click ‘Browse’
   3. Select ‘Rtutorial.ppk’
   4. Click ‘Open’ on the PuTTY screen
   5. Click ‘Open’ again
      1. Click: ‘Yes’ when a warning appears
7. PuTTY connects to my instance

The next step is to install FileZilla and connect FileZilla to the running AWS instance. FileZilla is used to transfer R and data files back and forth between your physical computer on your desk and your virtual AWS instance on the cloud.

1. Download FileZilla by going to this website
   1. <https://filezilla-project.org/>
   2. Click on ‘Download FileZilla Client All platforms’
   3. That takes me to this webpage:
      1. <https://filezilla-project.org/download.php?type=client>
      2. Download FileZilla Client for Windows (64bit x86)
2. I do not wish to reinstall FileZilla. Plus, I actually use FileZilla Pro which cost maybe $25.00.
   1. I do not recall how complex installing FileZilla was, but I suspect it was almost as easy as installing PuTTY
   2. I have located two YouTube videos that show how to install FileZilla on Windows 10
      1. The following video is straight-forward and shows exactly how to install FileZilla in approximately 4:30 minutes. Installation is easy and essentially the same as any typical Windows application
         1. <https://www.youtube.com/watch?v=cn4zTRxg-Oc>
3. In the ‘Host:’ box in the top left corner of FileZilla paste in the same string you used in PuTTY
   1. ubuntu@ec2-1-23-456-789.us-east-2.compute.amazonaws.com
4. In the ‘Port:’ box enter 22.
5. Click ‘QuickConnect’
6. FileZilla might fail to connect to the instance if FileZilla is not aware of the new key pair
7. Provide FileZilla with the new key pair using the following steps
   1. Click ‘Edit > Settings > SFTP’
   2. Click ‘Add key file…’
   3. Browse to the new key pair in ppk format and select it
   4. Click ‘Open’
   5. Click ‘OK’
   6. This answer was provided by rohit dixit
      1. https://superuser.com/questions/852066/no-supported-authentication-methods-available-in-filezilla
8. Browse to the R files and folders created above on your physical computer in the box in the lower left corner of FileZilla.

The above steps show how to install AWS, PuTTY and FileZilla and connect the latter two to a running instance in AWS. I could proceed here showing how to run the R files in either the folder ‘R\_files\_a’ or ‘R\_files\_b’. With only ten small R files in each folder this would be efficient. However, the ultimate objective is to demonstrate running a large number of R files in two different folders at the same time using Spot Instances. So, having learned the above steps it is more practical to terminate the existing running instance in AWS and create multiple spot instances using the security group and key pair we created above.

1. Close FileZilla
2. Close PuTTY
   1. When PuTTY asks, ‘Are you sure you want to close this session?’
   2. Click ‘OK’
3. To terminate the running instance in AWS:
   1. At the top of the screen next to the orange ‘Launch instances’ button click ‘Instance state > Terminate instance’
   2. Click the orange ‘Terminate’ button
   3. The ‘Instance state’ should switch to ‘Shutting-down’ and then ‘Terminated’ in red font

Now we are ready to set up our Spot Instances in AWS.

1. Click ‘Spot Requests’ in the left-hand column in AWS
2. Click ‘Request Spot Instances’
3. On the ‘Request Spot Instances’ screen select: ‘Flexible workloads’
4. Keep the default setting for ‘Launch template’: None
   1. Actually this is the only option apparently
5. For AMI select:
   1. Ubuntu Server 18.04 LTS (HVM), SSD Volume Type
6. For ‘Minimum compute unit’ keep the default: ‘as an instance type’
7. Click: Change Instance Type
   1. Click: t2.micro
      1. Note that this instance type might cost $0.0035 per hour
   2. Click the orange ‘Select’ button
8. Under ‘Network’ keep the default
9. Under ‘Availability Zone’
   1. Keep the default
      1. ‘No preference (balance across all Availability Zone)’
10. Under ‘Key pair name’
    1. Select: Rtutorial
11. Under ‘Total target capacity’
    1. Select: 2 instances
12. Click the box to the left of ‘Maintain target capacity’
    1. For ‘Interruption behavior’ select ‘Stop’
13. Click the box to the left of ‘Capacity rebalance’
14. A ‘fleet’ of possible instance types will be displayed along with the hourly rate for each. As far as I know the instance types eventually assigned to me are draw from this list with no further say on my part.
    1. I have always gotten this message under the list of instance types: ‘Fleet strength: Strong’
15. Under ‘Additional request details’
    1. Unclick the box next to ‘Apply defaults’
    2. Under ‘Request valid until’
       1. I set the instance to terminate after one day (24 hours)
       2. This is so if I forget to terminate the instance I will not be billed until I next log into AWS and discover the instance still running potentially months into the future.
16. My fleet request at a glance
    1. Total target capacity
       1. 2 instances
    2. Instance configuration:
       1. Custom
       2. 1 vCPU, 1 GiB (min) | 3 Availability Zones
    3. Fleet strength
       1. Strong
       2. 27 instance pools
    4. Estimated price
       1. ~$0.016/hr

at target capacity

* + 1. 69% savings

compared to On-Demand

1. Click ‘Launch’
2. Click ‘Instances’ in the left-hand column
   1. I have two t3.micro instances running
3. Click the box next to the Instance ID of one instance.
   1. Change the security group for this instance to: Rtutorial
      1. Click: Actions > Security > Change security groups
         1. Click inside the box next to ‘Associated security groups’
         2. Select: Rtutorial
         3. Click: ‘Add Security Group’
         4. Click ‘Remove’ next to the ‘default’ security group
         5. Click the orange ‘Save’ button
      2. Repeat this process for the second instance
      3. If you click the ‘Security’ tab under the list of running instances you will see which security group is assigned to a given instance
4. Under the ‘Details’ tab copy the ‘Public IPv4 DNS’ string for each instance
   1. Instance 1
      1. ec2-1-234-567-890.us-east-2.compute.amazonaws.com
      2. The actual numbers in this string are fake and are for illustration
   2. Instance 2
      1. ec2-2-234-567-890.us-east-2.compute.amazonaws.com
      2. The actual numbers in this string are fake and are for illustration
5. Open PuTTY
   1. Follow the instructions for PuTTY provide above and summarized here.
   2. Recall that you do not need to convert the format of the key pair. Reformatting is only necessary once as long as you use that key pair.
   3. For the first instance use:
      1. Host: [ubuntu@1-234-567-890.us-east-2.compute.amazonaws.com](mailto:ubuntu@1-234-567-890.us-east-2.compute.amazonaws.com)
      2. Connection: 180 seconds
      3. SSH
         1. Auth
            1. Browse
            2. Select key: Rtutorial.ppk
      4. Click: Open on the PuTTY screen
      5. Click: ‘Yes’ if a warning appears
      6. PuTTY connected
6. Open FileZilla
   1. Host: [ubuntu@ec2-3-133-141-240.us-east-2.compute.amazonaws.com](mailto:ubuntu@ec2-3-133-141-240.us-east-2.compute.amazonaws.com)
   2. Port: 22
   3. Click: Quickconnect
7. In PuTTY:
   1. Install R using these four lines one at a time. Note that if you copy and paste lines from MSWord directly into PuTTY an error could result because MSWord automatically places an outline designation immediately to the left of each line. One solution is to copy lines from MSWord and paste them into Notepad. There remove the outline designation from each line. Then copy the lines and paste them into PuTTY one at a time. However, as a more efficient solution I have left an empty space at the beginning of each of the four lines below.
      1. sudo su
      2. echo "deb [http://cran.rstudio.com/bin/linux/ubuntu trusty/](http://cran.rstudio.com/bin/linux/ubuntu%20trusty/)"
      3. sudo apt-get update
      4. sudo apt-get install r-base
         1. After the fourth line above a message will appear stating that additional disk space will be used.
         2. You will be asked ‘Do you want to continue? [Y/n]’
            1. Type Y
            2. Press ‘Enter’
      5. To open R type R and press ‘Enter’
      6. To quit R type q() and press ‘Enter’
         1. When asked ‘Save workspace image?’ type n and press ‘Enter’
8. Install dos2unix using:
   1. sudo apt install dos2unix
9. Install the R package doParallel
   1. Open R
   2. Use this line inside R:
      1. install.packages('doParallel')
   3. Close R: q()
      1. Type n and press ‘Enter’
10. Copy these files and folders from your computer to the home/ubuntu directory using FileZilla
    1. The R\_files\_a folder
       1. Click this folder in the lower-left window of FileZilla
       2. Drag and drop the folder into the lower-right window of FileZilla
    2. The Input\_data\_files\_a folder
    3. The Output\_data\_files\_a folder
    4. master\_R\_file\_a.R
    5. bash\_file\_a.sh
11. To submit the bash file paste these four lines in PuTTY one at a time:
    1. dos2unix bash\_file\_a.sh
    2. chmod +x bash\_file\_a.sh
    3. ./bash\_file\_a.sh &
    4. disown
12. You could close PuTTY and FileZilla
    1. However, the ten output files in this example will be created almost instantaneously.
13. Rename or delete the Output\_data\_files\_a folder from your physical computer and copy this folder from the virtual computer using FileZilla. Paste the output folder onto your physical computer using the same click, drag and drop procedure used to put it onto the virtual computer above.
14. A quick visual inspection of the ten output CSV files reveals all are correct.
15. Close FileZilla and PuTTY
16. Repeat Steps 47 – 51 for the second instance
    1. Host ubuntu@ec2-2-234-567-890.us-east-2.compute.amazonaws.com
17. Copy these files and folders from your computer to the home/ubuntu directory using FileZilla
    1. The R\_files\_b folder
       1. Click this folder in the lower-left window of FileZilla
       2. Drag and drop the folder into the lower-right window of FileZilla
    2. The Input\_data\_files\_b folder
    3. The Output\_data\_files\_b folder
    4. master\_R\_file\_b.R
    5. bash\_file\_b.sh
18. To submit the bash file paste these four lines in PuTTY one at a time:
    1. dos2unix bash\_file\_b.sh
    2. chmod +x bash\_file\_b.sh
    3. ./bash\_file\_b.sh &
    4. disown
19. Rename or delete the Output\_data\_files\_b folder from your physical computer and copy this folder from the virtual computer using FileZilla. Paste the output folder onto your physical computer using the same click, drag and drop procedure used to put it onto the virtual computer above.
20. A quick visual inspection of the ten output CSV files reveals all are correct.
21. Close FileZilla and PuTTY
22. Terminate the two instances in AWS
23. Click ‘Spot Requests’
    1. Cancel the fleet request by:
       1. Clicking the square box to the left of the Request ID
       2. Click Actions > Cancel Request
       3. Click ‘Confirm’
    2. The State of the fleet will change to ‘cancelled’ in red font
24. To view how much money you spent wait approximately 24 hours
    1. Click your name in the upper-right corner of the AWS windoe
       1. Click: My Billing Dashboard

Repeating steps 47 – 51 above is tedious. There seems to be a way around that but might require creating a custom AMI. That topic is something I might return to in the near future.