# Installing and using R on AWS

Kim Dill-McFarland, kadm@uw.edu

version September 23, 2021

This is a bare bones tutorial of how to install R and associated packages on an Amazon AMI EC2 instance. It assumes that you have followed AWS setup tutorial to create an AWS account and are familiar with the steps to create an EC2 instance.

## Create EC2 instance

Online through AWS console, create an instance (https://us-west-2.console.aws.amazon.com/ec2).

Recommendations for R usage:

- Amazon Linux 2 AMI
- Memory optimized type because R is a RAM hog
  - "m5" and "r5" types recommended
- Increase base EBS volume to 32 GB to hold programs and packages
- Add additional 1TB EBS volume(s) up to the amount you will need for all your raw and processed data

## Basic setup

In the Terminal, log-in into your instance and perform basic setup. More details are available in `1.AWS_setup_tutorial.pdf`.

This directory setup is slightly different than in the setup tutorial. We mount the EBS storage to `project/` and then fuse a subdirectory `project/data/` to our S3 bucket holding all our data. This is because a fused directory becomes *read-only*. Thus, if we fused to the main directory as in the setup tutorial, we would not be able to write any of our results to the EBS storage where we have the extra space.

```
## Updates if available
sudo yum upgrade -y
sudo yum update -y

## Install AWS command line client if not using Amazon OS
sudo yum install awscli -y

## Configure your account
aws configure
## FILL IN WITH YOUR KEYS ###

## Setup fuse
sudo amazon-linux-extras install -y epel
sudo yum install -y s3fs-fuse
### Fuse key
### FILL IN WITH YOUR KEYS ###
echo UserKey:SecretKey > ~/.passwd-s3fs
```

```
chmod 600  ~/.passwd-s3fs

## Setup EBS volumes
lsblk

sudo mkfs -t ext4 /dev/nvme1n1
sudo mkdir -p ~/project
sudo mount /dev/nvme1n1 ~/project/
### Change permissions
sudo chmod 777 -R ~/project/

## Mount S3 data
mkdir ~/project/data
sudo chmod 777 -R ~/project/data

s3fs kadm-data ~/project/data -o passwd_file=~/.passwd-s3fs \
    -o default_acl=public-read -o uid=1000 -o gid=1000 -o umask=0007
```

# Install R

## Download R

Here, we download and install R on the main EC2 instance storage, not EBS extra storage. You could install it to extra storage if you main disk was not large enough.

```
### Download to a directory on the main disk
mkdir ~/apps/
sudo chmod 777 -R ~/apps
cd ~/apps/
#### Update to latest version as necessary ####
wget https://cran.r-project.org/src/base/R-4/R-4.1.1.tar.gz
tar xf R-4.1.1.tar.gz
cd R-4.1.1/
```

## Install dependencies

```
sudo yum install -y gcc gcc-c++ gcc-gfortran readline-devel \
  zlib-devel bzip2 bzip2-devel xz xz-devel \
  libcurl libcurl.i686 libcurl-devel.x86_64 \
  openssl-devel findutils libffi-devel \
  libxml2-devel PCRE java

sudo yum update -y
```

## Compile and install R

**This will take several minutes.**

```
./configure --prefix=$HOME/R-4.1.1/ --with-x=no --with-pcre1
make
```

### Set default path to R

Add the PATH to R to `~/.bash_profile`. Once complete, exit and re-login to your EC2 instance for the PATH to take effect.

```
echo export PATH=~/apps/R-4.1.1/bin:$PATH >> ~/.bash_profile
```

## Install packages

Open R with `R` [Enter]. Double check that the version that opens is the one you downloaded. If it is not, there is likely something wrong with your PATH in `.bash_profile`.

Install packages as you would in RStudio's console. You can speed thing up by setting the number of threads (or CPUs) in options like so. The following are recommended to install on all EC2 instances with R.

```
options(Ncpus = 20)

install.packages(c("foreach","doParallel",
                   "tidyverse","BiocManager","devtools"),
  repos='http://cran.us.r-project.org')
```

Exit with the following.

```
q()
```

## Run a script

In the terminal, you can run scripts like so. I recommend using `screen` so that the script runs even if you log-out of the instance.

```
screen

Rscript my_script.R
```

At any time, you can check what's running on your instance with `top` in the terminal.

---