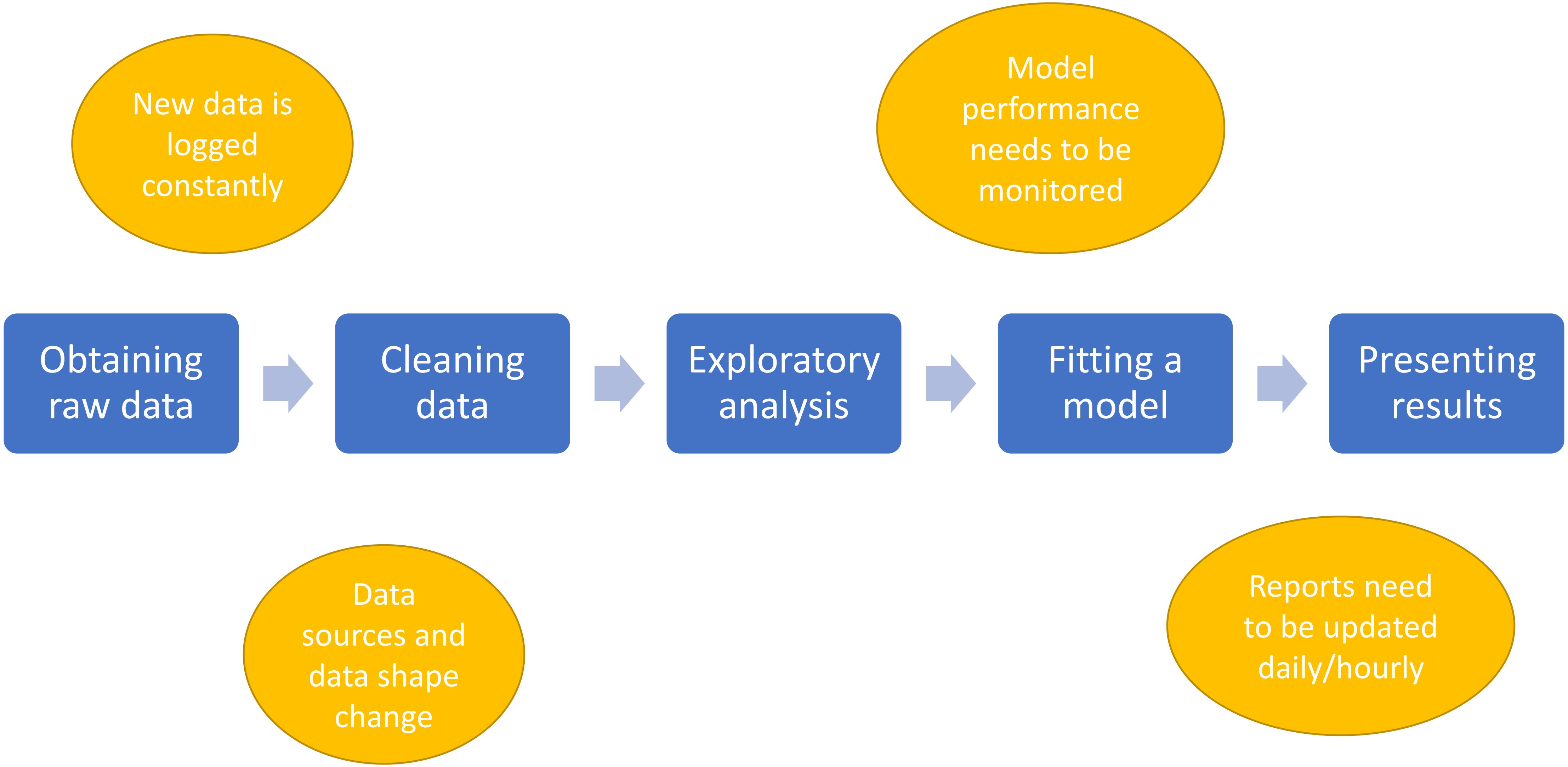
From laptop to cloud: running R on AWS

Ksenia Aleksankina

travelnest



Obtaining raw data

Cleaning data

Exploratory analysis

Fitting a model

Presenting results

New data is logged constantly

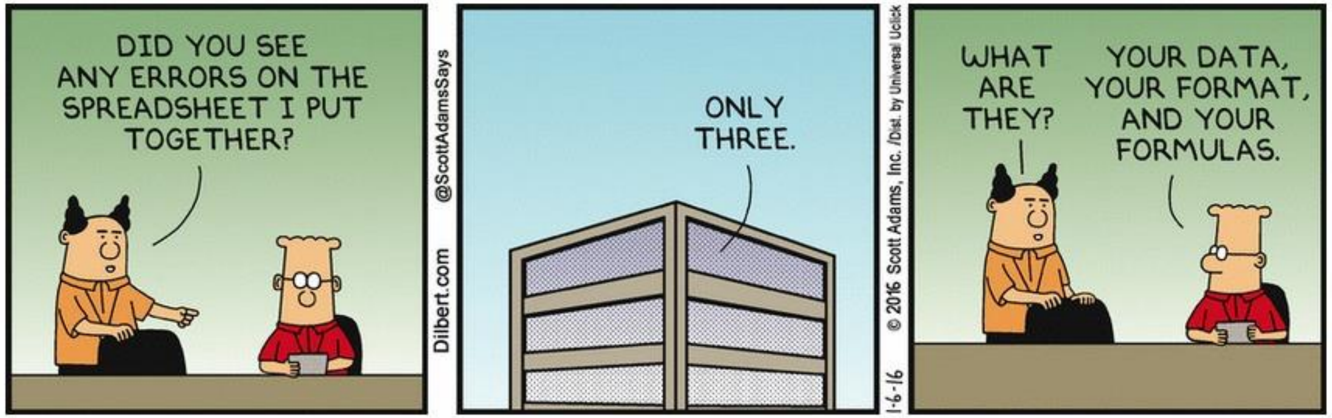
Data sources and data shape change

Model performance needs to be monitored

Reports need to be updated daily/hourly

Some of the resulting issues:

- Difficulties in consuming data
- Reproducibility
- Difficulties in communicating analysis results


DID YOU SEE ANY ERRORS ON THE SPREADSHEET I PUT TOGETHER?
ONLY THREE.
WHAT ARE THEY?
YOUR DATA, YOUR FORMAT, AND YOUR FORMULAS.

Dilbert.com @ScottAdamsSays

1-6-16 © 2016 Scott Adams, Inc. /Dist. by Universal Uclick

Solution:

- Deploy and access R/RStudio/Shiny on a cloud server
- Collect, store, and pre-process data in the cloud
- Keep analysis results or model predictions up to date and accessible

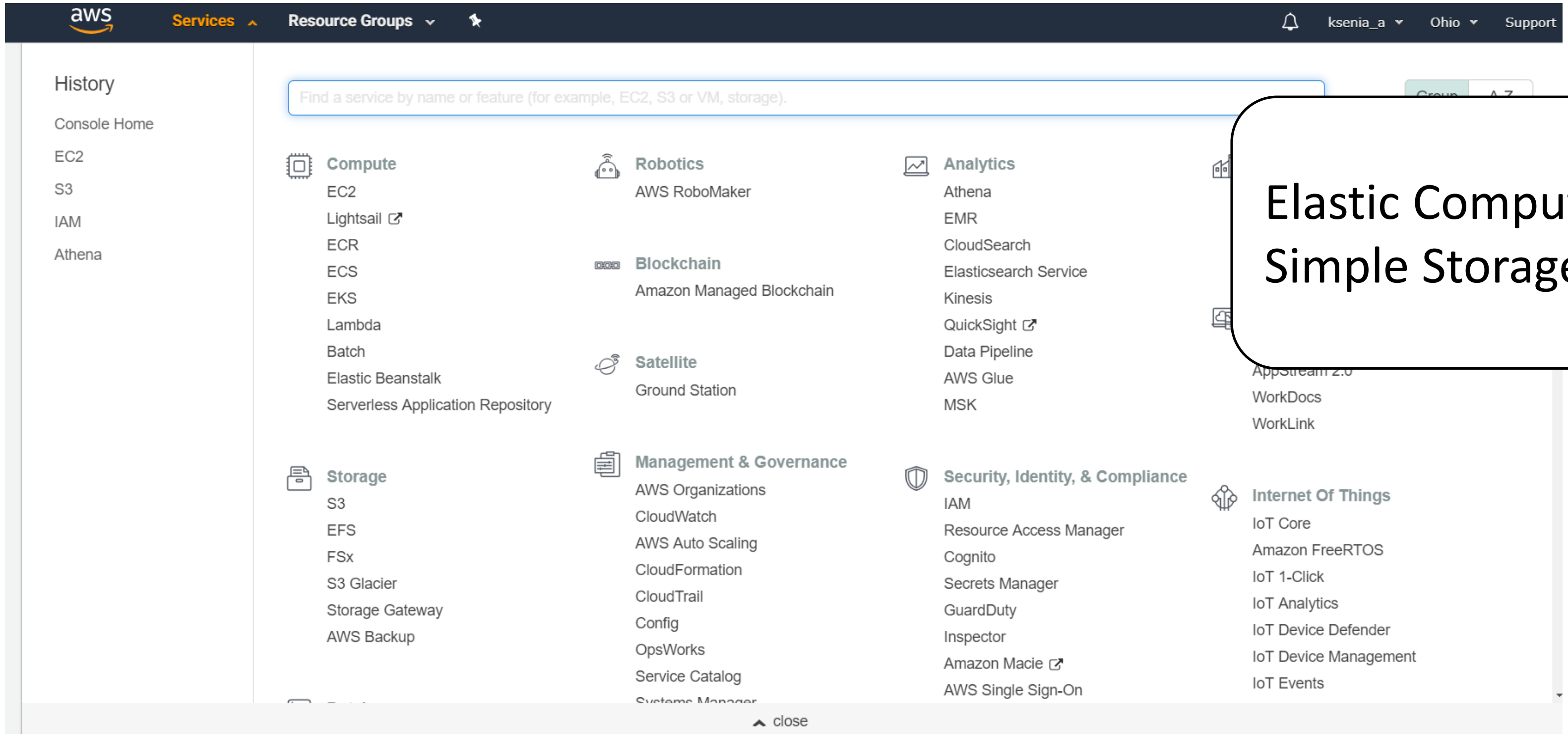
Why AWS?

Amazon Web Services is an on-demand cloud computing platform

- Variety of building blocks and structures
- Pay-as-you-go (with a free tier)

amazon web services & R

The account is created, now what?

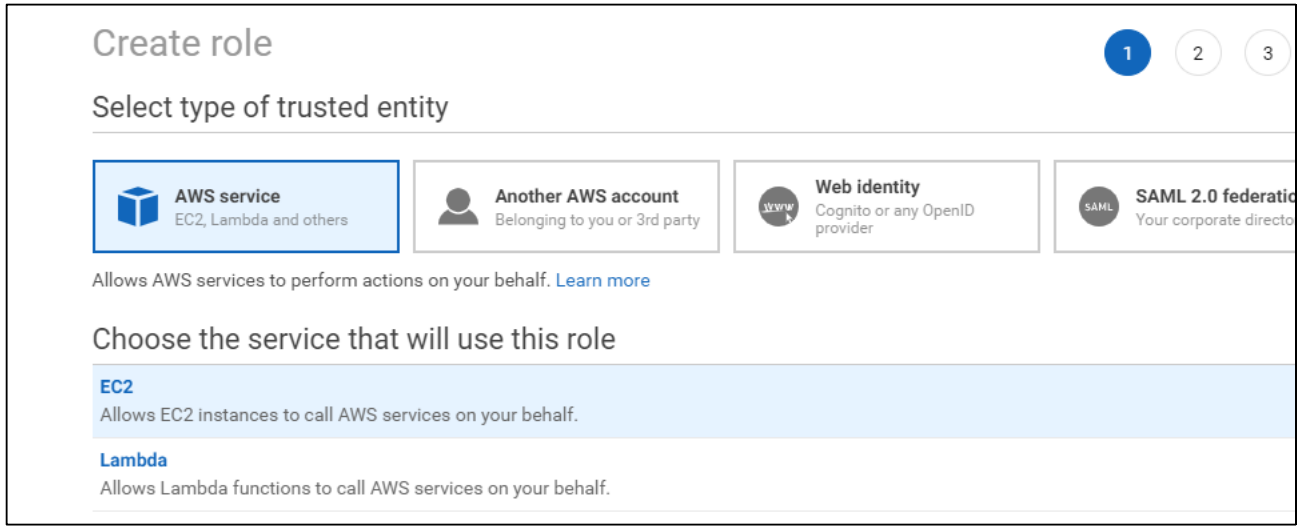


Elastic Compute Cloud (EC2)
Simple Storage Service (S3)

https://aws.amazon.com/blogs/big-data/running-r-on-aws/
http://stanke.co/r-on-aws-cloud/

Before launching EC2 instance

Create an Identity and Access Management (IAM) role



Create role

Select type of trusted entity

Allows AWS services to perform actions on your behalf. Learn more

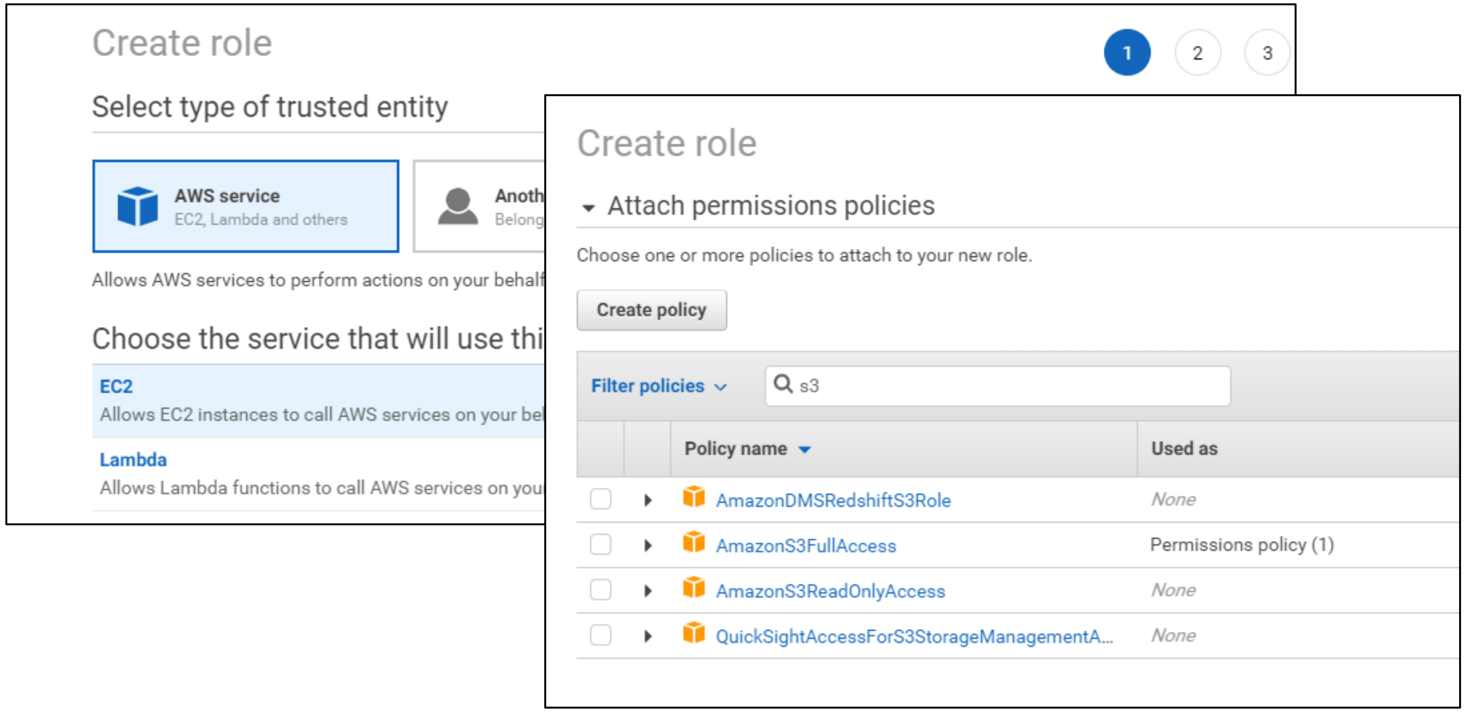
Choose the service that will use this role

EC2
Allows EC2 instances to call AWS services on your behalf.

Lambda
Allows Lambda functions to call AWS services on your behalf.

Before launching EC2 instance

Create an Identity and Access Management (IAM) role



Create role

Select type of trusted entity

Allows AWS services to perform actions on your behalf

Choose the service that will use thi

EC2
Allows EC2 instances to call AWS services on your be

Lambda
Allows Lambda functions to call AWS services on you

Create role

Attach permissions policies

Choose one or more policies to attach to your new role.

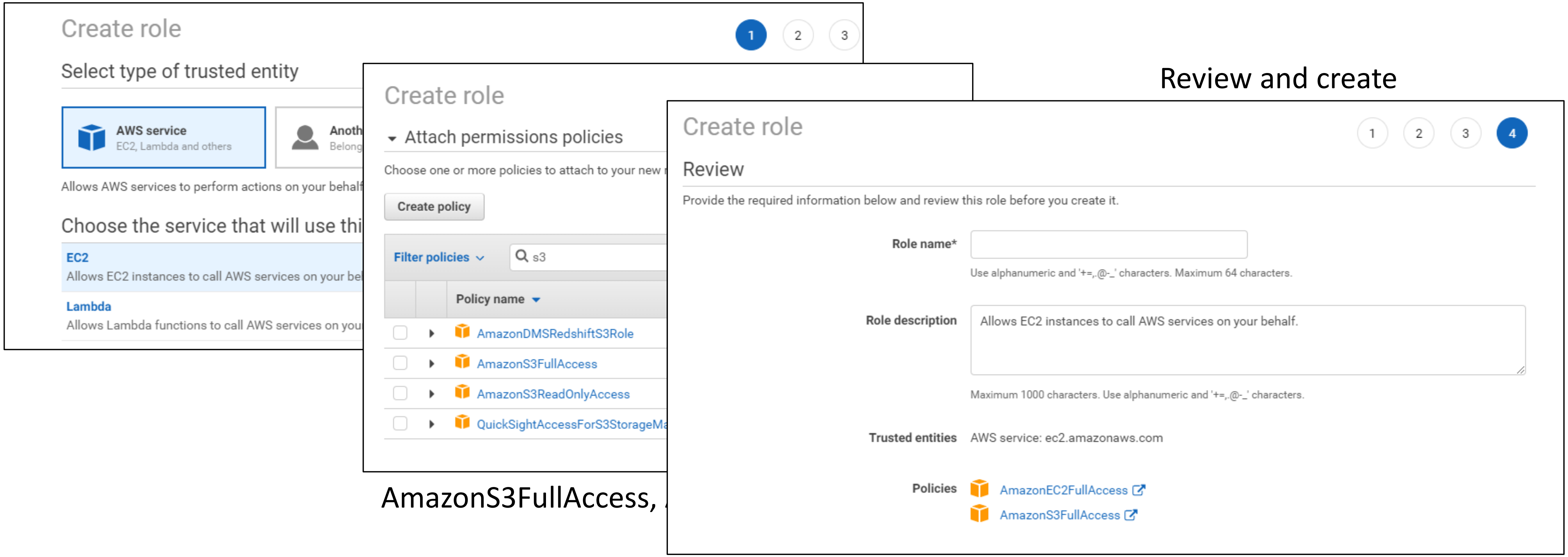
s3

		Policy name	Used as
□	▸	AmazonDMSRedshiftS3Role	*None*
□	▸	AmazonS3FullAccess	Permissions policy (1)
□	▸	AmazonS3ReadOnlyAccess	*None*
□	▸	QuickSightAccessForS3StorageManagementA...	*None*

AmazonS3FullAccess, AmazonEC2FullAccess

Before launching EC2 instance

Create an Identity and Access Management (IAM) role



Review and create

Create role

1 2 3

Select type of trusted entity

AWS service
EC2, Lambda and others

Anoth
Belong

Allows AWS services to perform actions on your behalf

Choose the service that will use thi

EC2
Allows EC2 instances to call AWS services on your be

Lambda
Allows Lambda functions to call AWS services on you

Create role

Attach permissions policies

Choose one or more policies to attach to your new r

s3

		Policy name
□	▸	AmazonDMSRedshiftS3Role
□	▸	AmazonS3FullAccess
□	▸	AmazonS3ReadOnlyAccess
□	▸	QuickSightAccessForS3StorageMa

Create role

1 2 3 4

Review

Provide the required information below and review this role before you create it.

Role name*

Use alphanumeric and '+=,.@-_' characters. Maximum 64 characters.

Role description: Allows EC2 instances to call AWS services on your behalf.

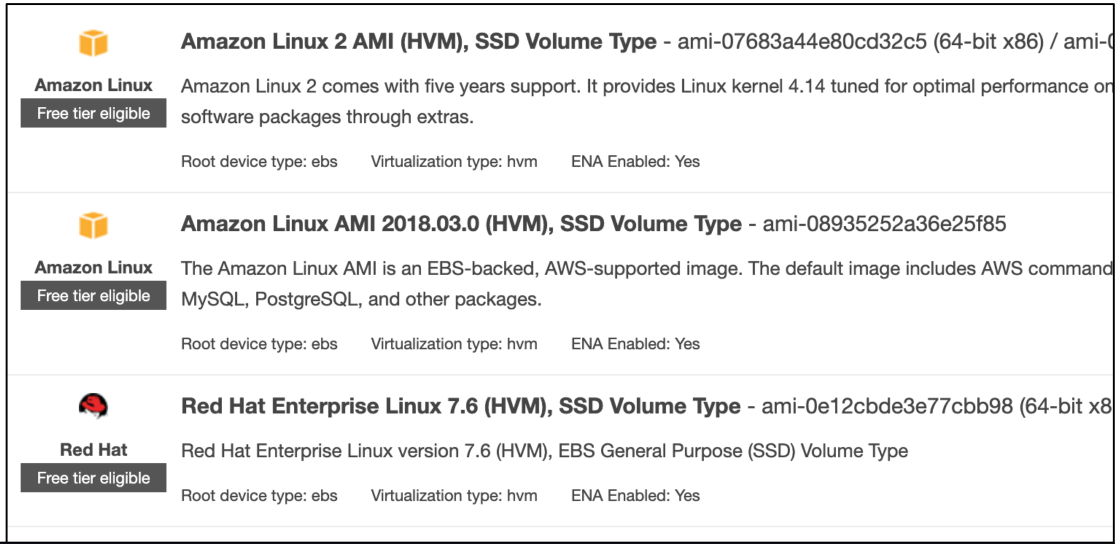
Maximum 1000 characters. Use alphanumeric and '+=,.@-_' characters.

Trusted entities: AWS service: ec2.amazonaws.com

Policies: AmazonEC2FullAccess
AmazonS3FullAccess

AmazonS3FullAccess,

- Chose an EC2 instance (AMI)
- Instance type (R runs only on one CPU but requires a lot of memory)



Amazon Linux
Free tier eligible

Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-07683a44e80cd32c5 (64-bit x86) / ami-(

Amazon Linux 2 comes with five years support. It provides Linux kernel 4.14 tuned for optimal performance on software packages through extras.

Root device type: ebs   Virtualization type: hvm   ENA Enabled: Yes

Amazon Linux
Free tier eligible

Amazon Linux AMI 2018.03.0 (HVM), SSD Volume Type - ami-08935252a36e25f85

The Amazon Linux AMI is an EBS-backed, AWS-supported image. The default image includes AWS command MySQL, PostgreSQL, and other packages.

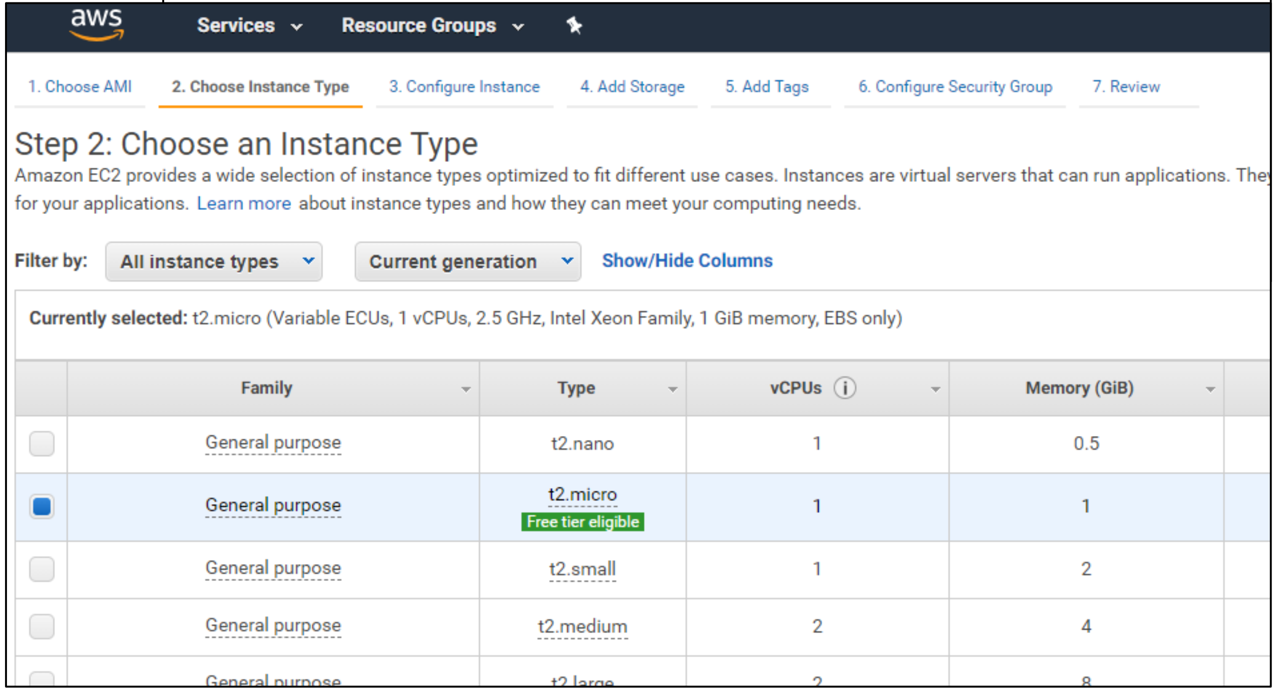
Root device type: ebs   Virtualization type: hvm   ENA Enabled: Yes

Red Hat
Free tier eligible

Red Hat Enterprise Linux 7.6 (HVM), SSD Volume Type - ami-0e12cbde3e77cbb98 (64-bit x8

Red Hat Enterprise Linux version 7.6 (HVM), EBS General Purpose (SSD) Volume Type

Root device type: ebs   Virtualization type: hvm   ENA Enabled: Yes



Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They for your applications. Learn more about instance types and how they can meet your computing needs.

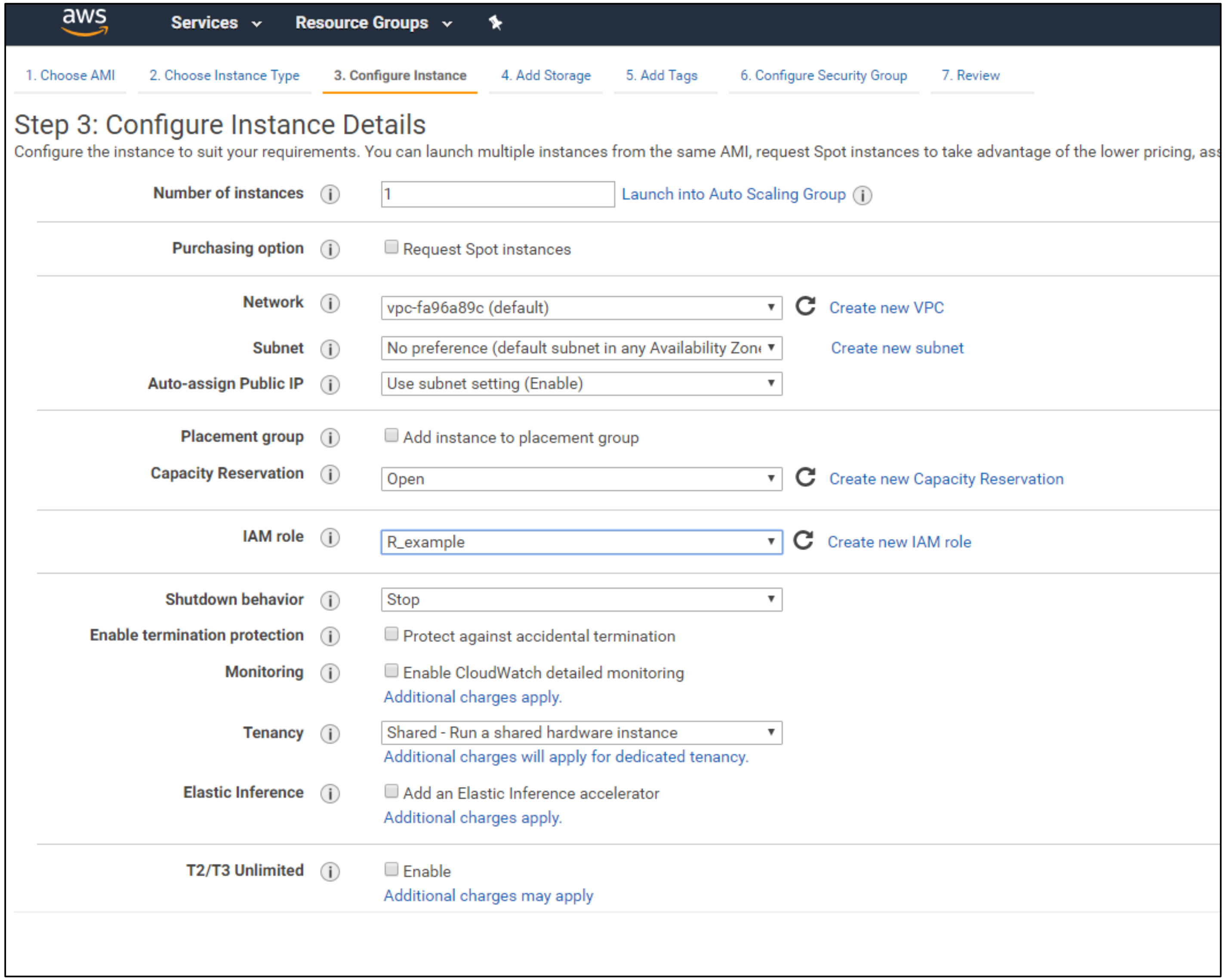
Filter by: All instance types   Current generation   Show/Hide Columns

Currently selected: t2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only)

	Family	Type	vCPUs	Memory (GiB)
□	General purpose	t2.nano	1	0.5
☑	General purpose	t2.micro Free tier eligible	1	1
□	General purpose	t2.small	1	2
□	General purpose	t2.medium	2	4
□	General purpose	t2.large	2	8

- Configure Instance

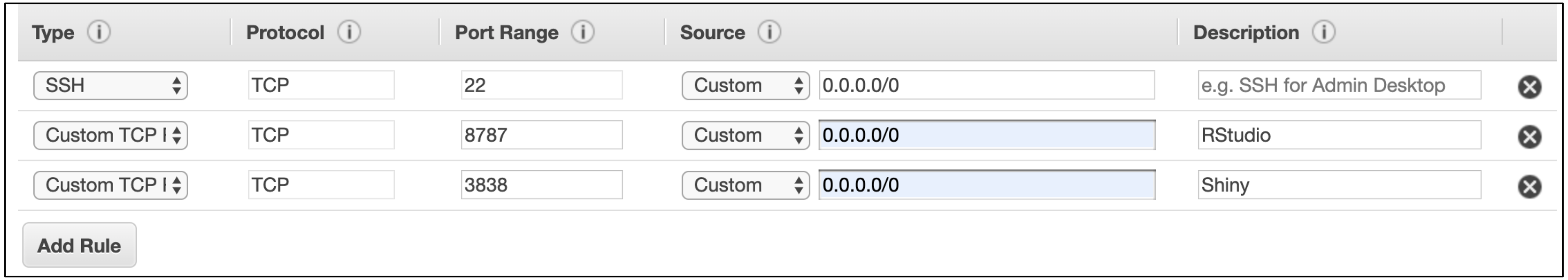
  Select previously created IAM



Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, ass

Number of instances	1 Launch into Auto Scaling Group
Purchasing option	□ Request Spot instances
Network	vpc-fa96a89c (default) Create new VPC
Subnet	No preference (default subnet in any Availability Zon Create new subnet
Auto-assign Public IP	Use subnet setting (Enable)
Placement group	□ Add instance to placement group
Capacity Reservation	Open Create new Capacity Reservation
IAM role	R_example Create new IAM role
Shutdown behavior	Stop
Enable termination protection	□ Protect against accidental termination
Monitoring	□ Enable CloudWatch detailed monitoring Additional charges apply.
Tenancy	Shared - Run a shared hardware instance Additional charges will apply for dedicated tenancy.
Elastic Inference	□ Add an Elastic Inference accelerator Additional charges apply.
T2/T3 Unlimited	□ Enable Additional charges may apply

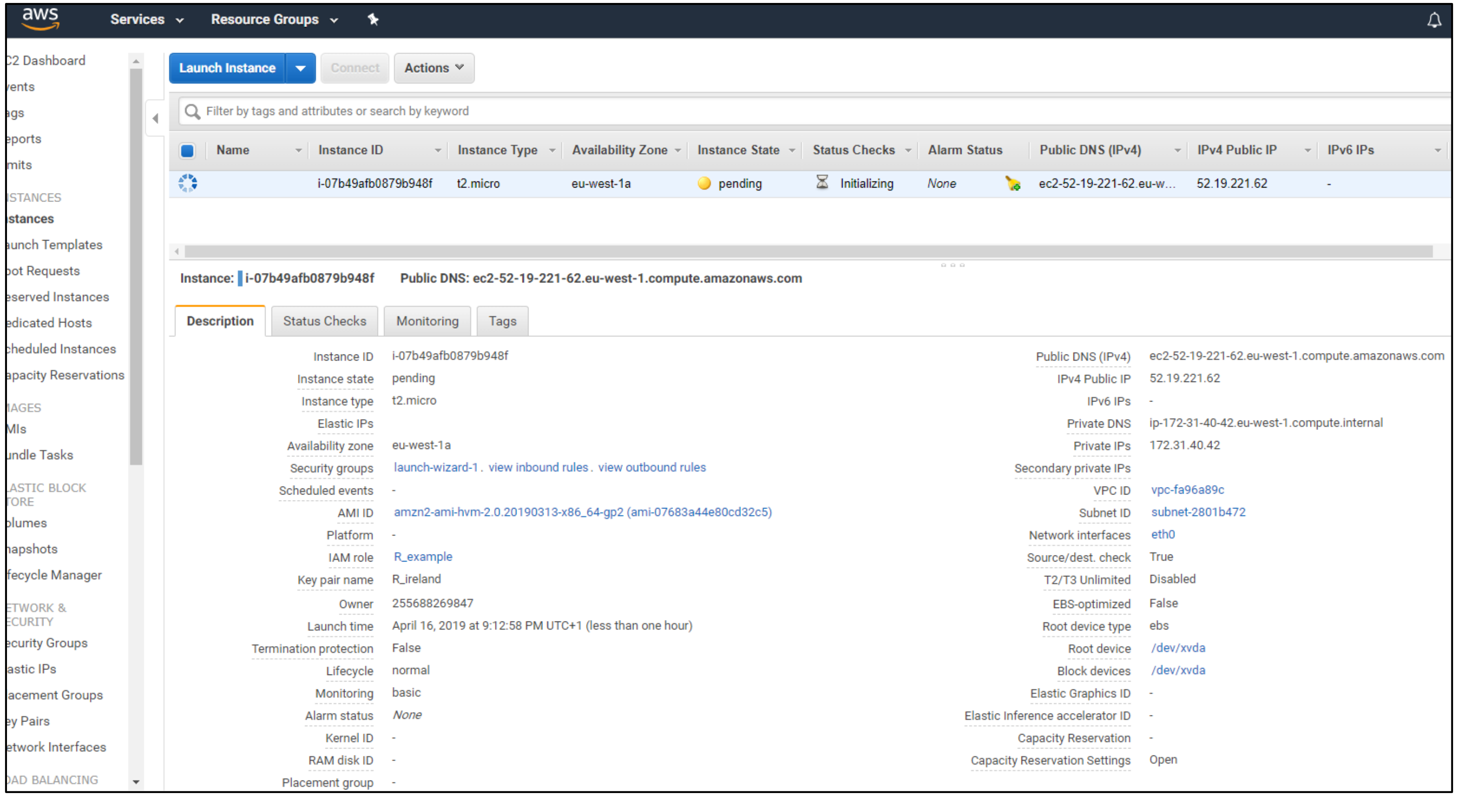


Type	Protocol	Port Range	Source		Description	
SSH	TCP	22	Custom	0.0.0.0/0	e.g. SSH for Admin Desktop	⊗
Custom TCP I	TCP	8787	Custom	0.0.0.0/0	RStudio	⊗
Custom TCP I	TCP	3838	Custom	0.0.0.0/0	Shiny	⊗

- Configure security groups

- 0.0.0.0/0 allow all IP addresses to access your instance

The instance is ready…



Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public IP	IPv6 IPs
	i-07b49afb0879b948f	t2.micro	eu-west-1a	● pending	⌛ Initializing	*None*	ec2-52-19-221-62.eu-w...	52.19.221.62	-

Instance: i-07b49afb0879b948f   Public DNS: ec2-52-19-221-62.eu-west-1.compute.amazonaws.com

Instance ID	i-07b49afb0879b948f
Instance state	pending
Instance type	t2.micro
Elastic IPs	
Availability zone	eu-west-1a
Security groups	launch-wizard-1 . view inbound rules . view outbound rules
Scheduled events	-
AMI ID	amzn2-ami-hvm-2.0.20190313-x86_64-gp2 (ami-07683a44e80cd32c5)
Platform	-
IAM role	R_example
Key pair name	R_ireland
Owner	255688269847
Launch time	April 16, 2019 at 9:12:58 PM UTC+1 (less than one hour)
Termination protection	False
Lifecycle	normal
Monitoring	basic
Alarm status	*None*
Kernel ID	-
RAM disk ID	-
Placement group	-

Public DNS (IPv4)	ec2-52-19-221-62.eu-west-1.compute.amazonaws.com
IPv4 Public IP	52.19.221.62
IPv6 IPs	-
Private DNS	ip-172-31-40-42.eu-west-1.compute.internal
Private IPs	172.31.40.42
Secondary private IPs	
VPC ID	vpc-fa96a89c
Subnet ID	subnet-2801b472
Network interfaces	eth0
Source/dest. check	True
T2/T3 Unlimited	Disabled
EBS-optimized	False
Root device type	ebs
Root device	/dev/xvda
Block devices	/dev/xvda
Elastic Graphics ID	-
Elastic Inference accelerator ID	-
Capacity Reservation	-
Capacity Reservation Settings	Open

… time to install R, Rstudio, and Shiny

```
# Install R
$ sudo yum install -y R

# Install RStudio Server
$ wget https://download2.rstudio.org/server/centos6/x86_64/rstudio-server-rhel-1.2.1335-x86_64.rpm
$ sudo yum install -y --nogpgcheck /rstudio-server-rhel-1.2.1335-x86_64.rpm

# Install Shiny and Shiny-server
$ R -e \"install.packages('shiny', repos='https://cran.rstudio.com/’)\”
$ wget https://download3.rstudio.org/centos6.3/x86_64/shiny-server-1.5.9.923-x86_64.rpm
$ yum install -y --nogpgcheck shiny-server-1.5.9.923-x86_64.rpm

# Add user(s)
$ sudo useradd –m *username*
$ sudo passwd *password*
```

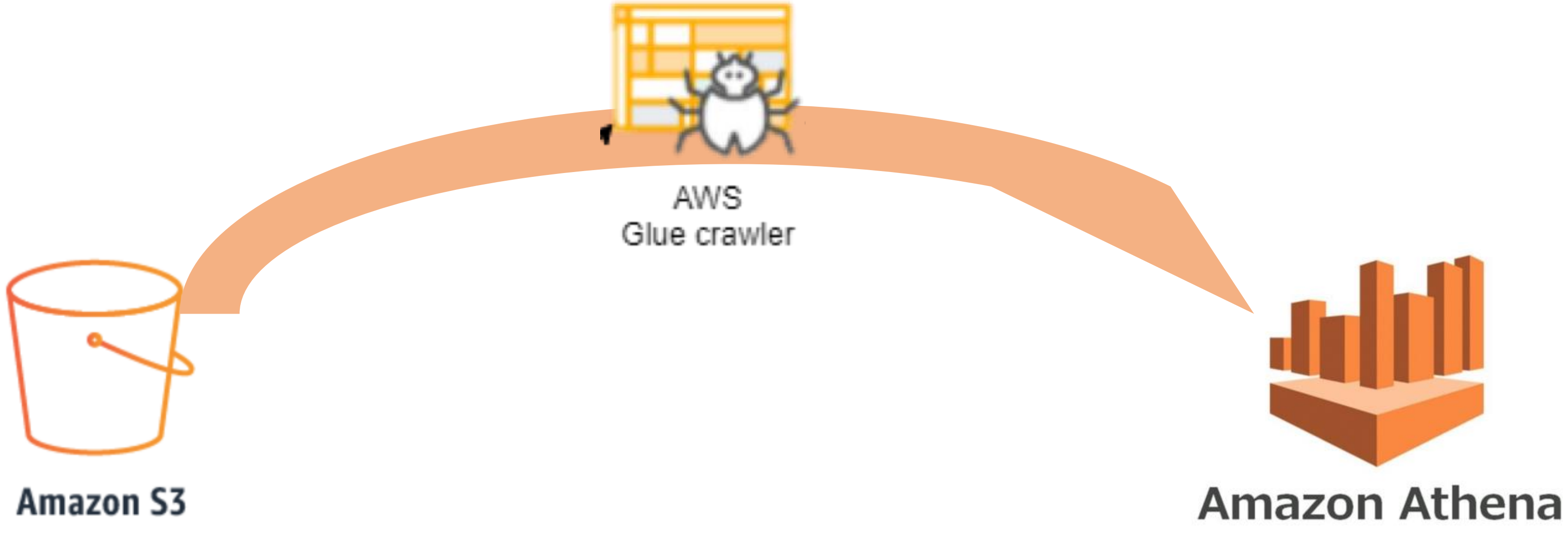
Issues along the way

- t2.micro did not like dplyr or leaflet

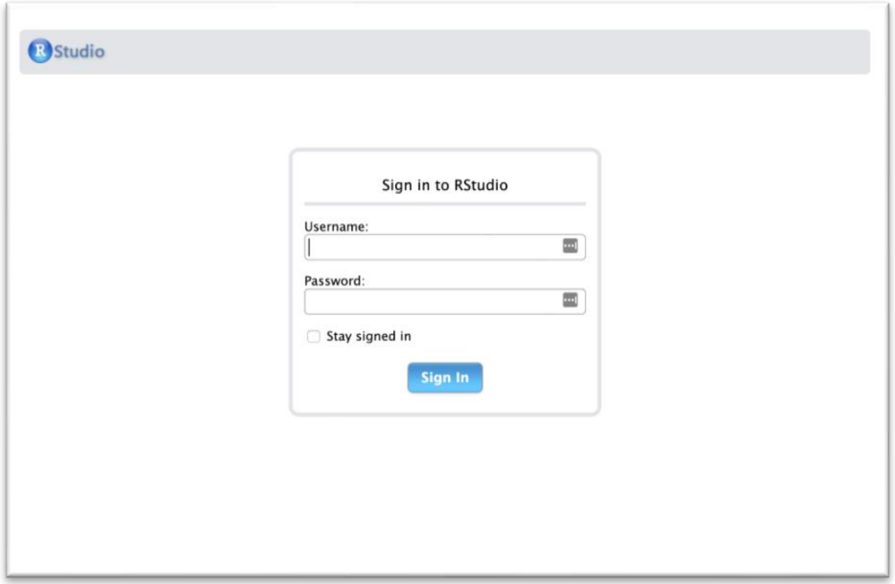
  Solution: change instance type to t2.medium
- Occasional ‘non-zero exit status’ errors
- For leaflet: Install Portable Network Graphics reference library

  ```
  $ sudo yum -y install libpng-devel
  ```

Simple Storage Service (S3)



RStudio



Sign in to RStudio

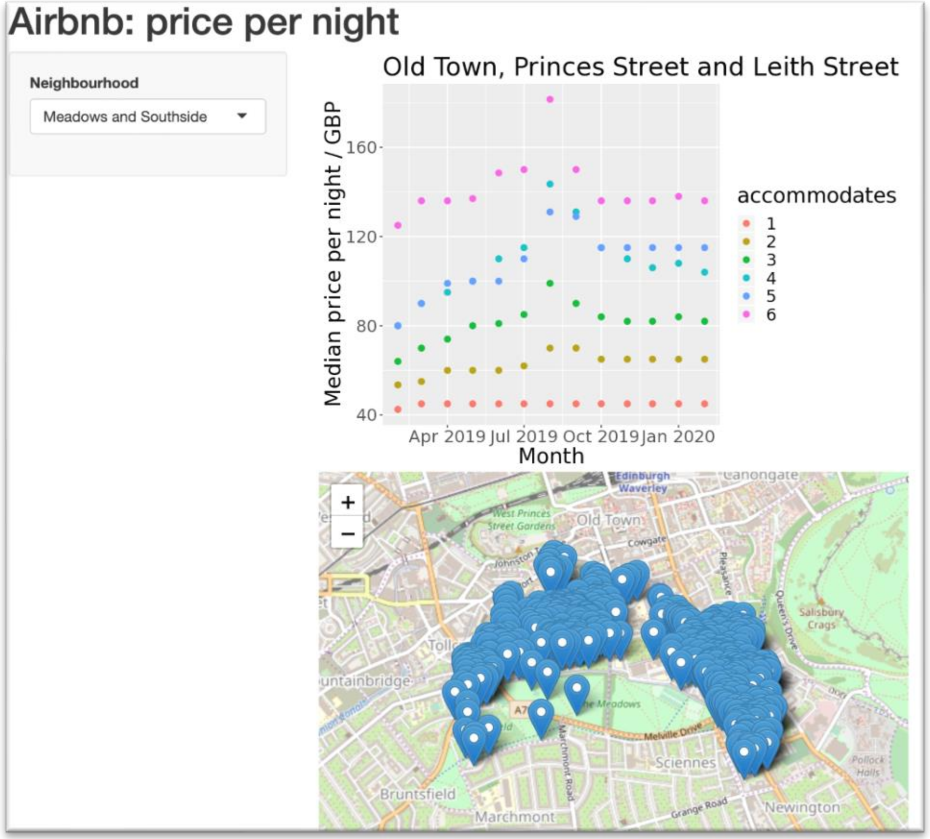
Username:

Password:

□ Stay signed in

Sign In

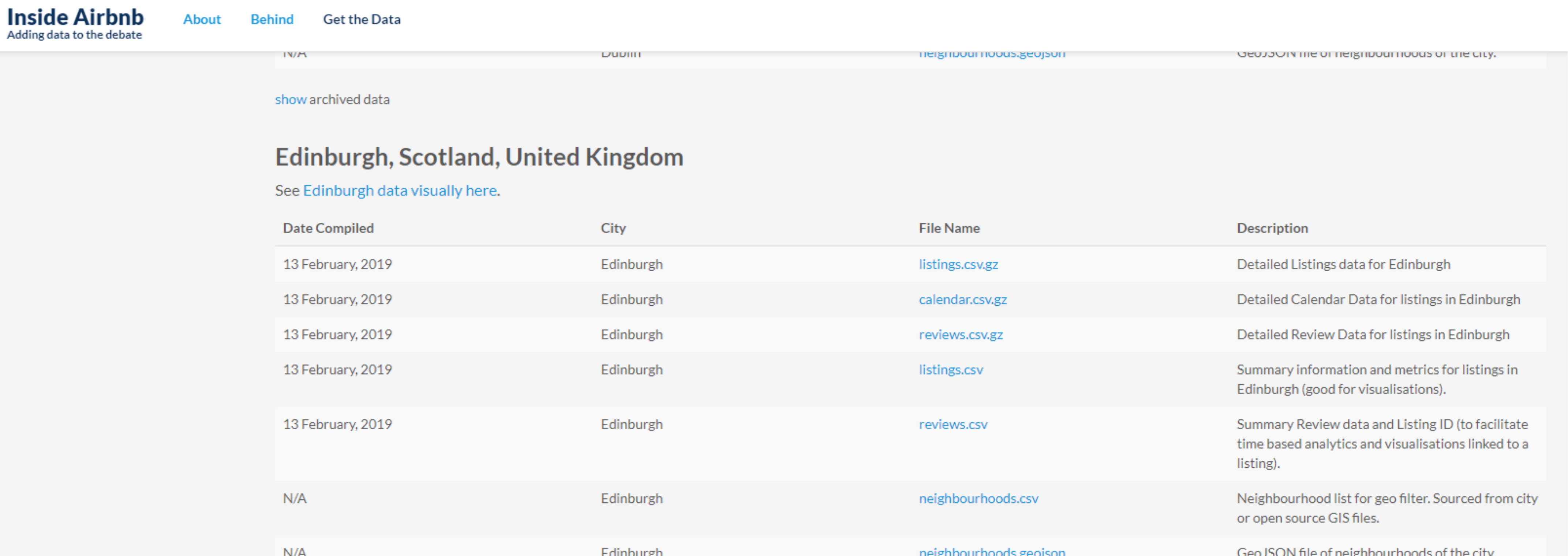
Shiny app



Airbnb: price per night

Neighbourhood: Meadows and Southside

Data for the Shiny app



N/A	Dublin	neighbourhoods.geojson	GeoJSON file of neighbourhoods of the city.

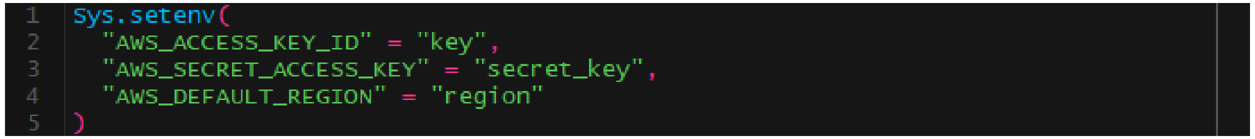
show archived data

Edinburgh, Scotland, United Kingdom

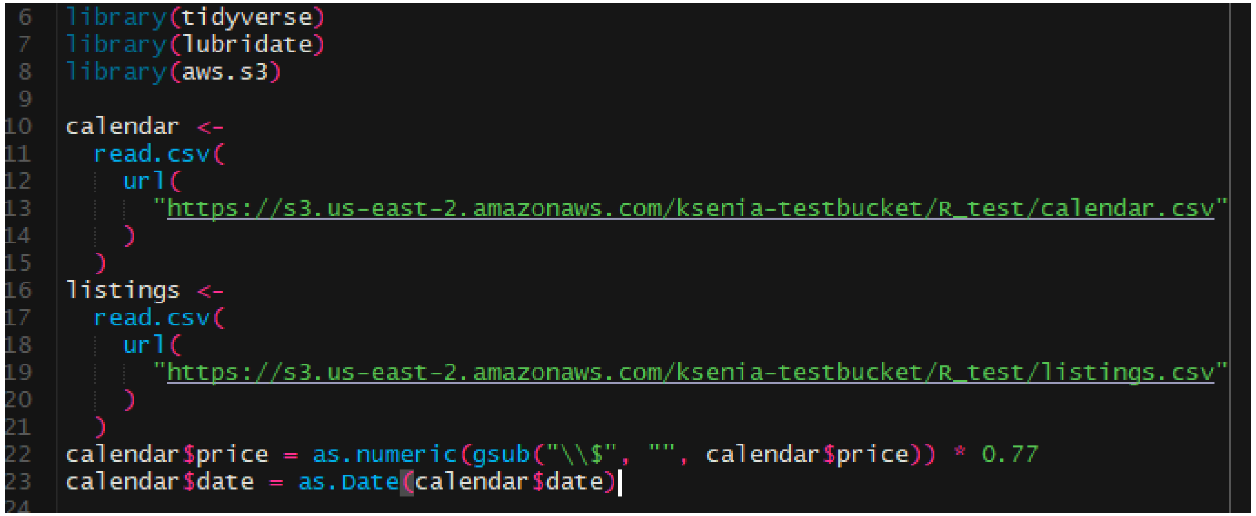
See Edinburgh data visually here.

Date Compiled	City	File Name	Description
13 February, 2019	Edinburgh	listings.csv.gz	Detailed Listings data for Edinburgh
13 February, 2019	Edinburgh	calendar.csv.gz	Detailed Calendar Data for listings in Edinburgh
13 February, 2019	Edinburgh	reviews.csv.gz	Detailed Review Data for listings in Edinburgh
13 February, 2019	Edinburgh	listings.csv	Summary information and metrics for listings in Edinburgh (good for visualisations).
13 February, 2019	Edinburgh	reviews.csv	Summary Review data and Listing ID (to facilitate time based analytics and visualisations linked to a listing).
N/A	Edinburgh	neighbourhoods.csv	Neighbourhood list for geo filter. Sourced from city or open source GIS files.
N/A	Edinburgh	neighbourhoods.geojson	GeoJSON file of neighbourhoods of the city.

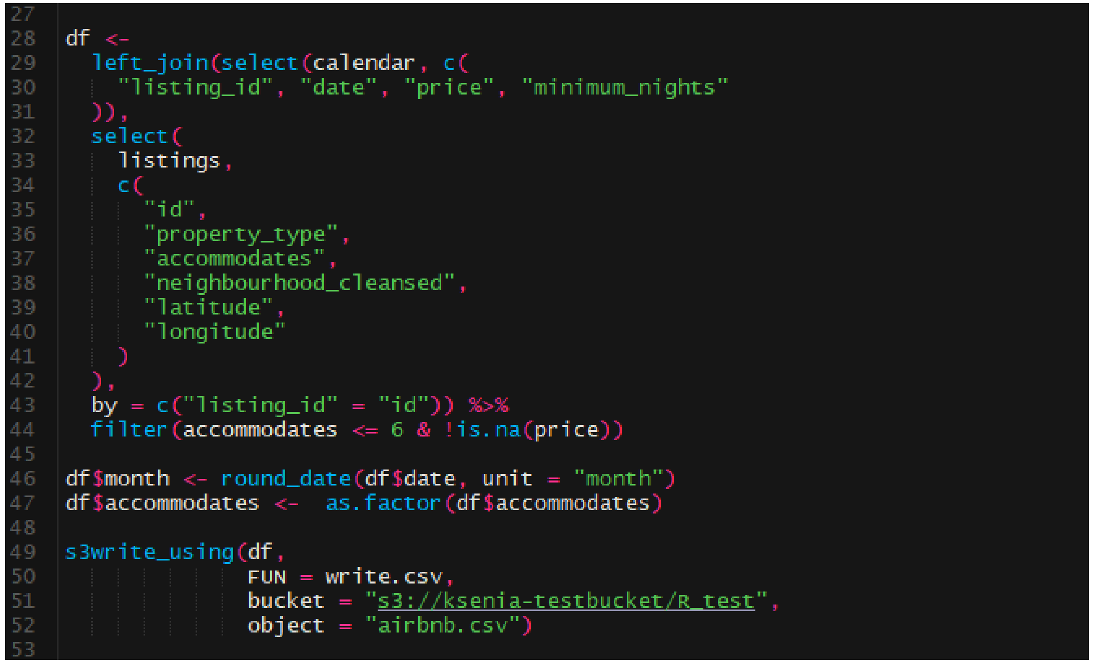
http://insideairbnb.com/about.html



```r
Sys.setenv(
  "AWS_ACCESS_KEY_ID" = "key",
  "AWS_SECRET_ACCESS_KEY" = "secret_key",
  "AWS_DEFAULT_REGION" = "region"
)
```

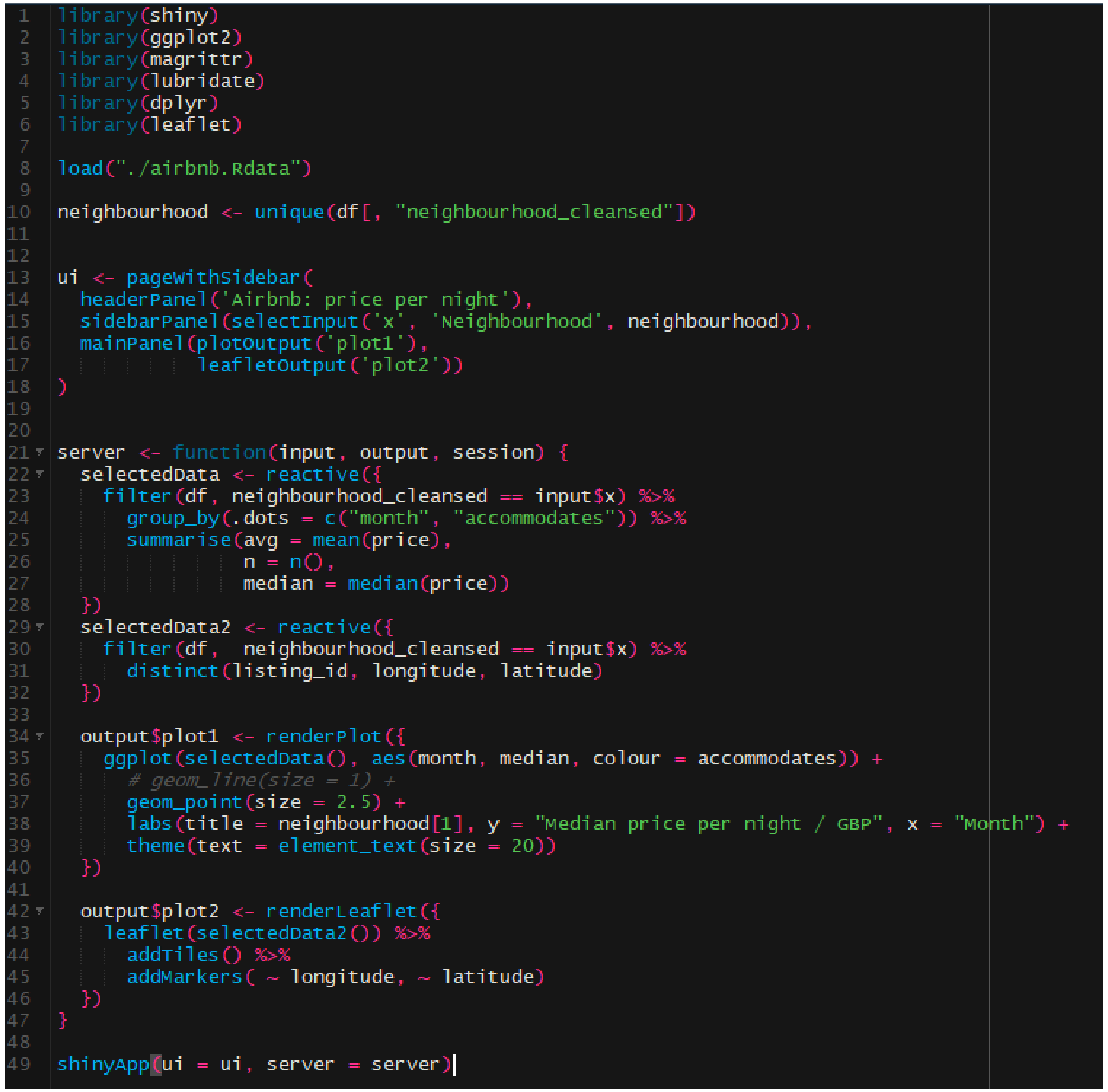


```r
library(tidyverse)
library(lubridate)
library(aws.s3)

calendar <-
  read.csv(
    url(
      "https://s3.us-east-2.amazonaws.com/ksenia-testbucket/R_test/calendar.csv"
    )
  )
listings <-
  read.csv(
    url(
      "https://s3.us-east-2.amazonaws.com/ksenia-testbucket/R_test/listings.csv"
    )
  )
calendar$price = as.numeric(gsub("\\$", "", calendar$price)) * 0.77
calendar$date = as.Date(calendar$date)
```



```r

df <-
  left_join(select(calendar, c(
    "listing_id", "date", "price", "minimum_nights"
  )),
  select(
    listings,
    c(
      "id",
      "property_type",
      "accommodates",
      "neighbourhood_cleansed",
      "latitude",
      "longitude"
    )
  ),
  by = c("listing_id" = "id")) %>%
  filter(accommodates <= 6 & !is.na(price))

df$month <- round_date(df$date, unit = "month")
df$accommodates <-  as.factor(df$accommodates)

s3write_using(df,
              FUN = write.csv,
              bucket = "s3://ksenia-testbucket/R_test",
              object = "airbnb.csv")

```

Shiny app code



```r
library(shiny)
library(ggplot2)
library(magrittr)
library(lubridate)
library(dplyr)
library(leaflet)

load("./airbnb.Rdata")

neighbourhood <- unique(df[, "neighbourhood_cleansed"])


ui <- pageWithSidebar(
  headerPanel('Airbnb: price per night'),
  sidebarPanel(selectInput('x', 'Neighbourhood', neighbourhood)),
  mainPanel(plotOutput('plot1'),
            leafletOutput('plot2'))
)


server <- function(input, output, session) {
  selectedData <- reactive({
    filter(df, neighbourhood_cleansed == input$x) %>%
      group_by(.dots = c("month", "accommodates")) %>%
      summarise(avg = mean(price),
                n = n(),
                median = median(price))
  })
  selectedData2 <- reactive({
    filter(df,  neighbourhood_cleansed == input$x) %>%
      distinct(listing_id, longitude, latitude)
  })

  output$plot1 <- renderPlot({
    ggplot(selectedData(), aes(month, median, colour = accommodates)) +
      # geom_line(size = 1) +
      geom_point(size = 2.5) +
      labs(title = neighbourhood[1], y = "Median price per night / GBP", x = "Month") +
      theme(text = element_text(size = 20))
  })

  output$plot2 <- renderLeaflet({
    leaflet(selectedData2()) %>%
      addTiles() %>%
      addMarkers( ~ longitude, ~ latitude)
  })
}

shinyApp(ui = ui, server = server)
```

Useful resources

- Cloudyr project http://cloudyr.github.io/
- Running R on AWS https://aws.amazon.com/blogs/big-data/running-r-on-aws/
- Taking Advanced Analytics to the Cloud http://stanke.co/r-on-aws-cloud/
- What Is Amazon S3? https://docs.aws.amazon.com/AmazonS3/latest/dev/Welcome.html
- An Introduction to Rocker: Docker Containers for R https://arxiv.org/abs/1710.03675

Thank you!