Pushing R Further

In the previous chapters we worked with data on a single machine and we presented a number of solutions to the traditional limitations of the R programming language: storage of data structures in memory and limited processing speed compared to compiled languages. Of course, a single-machine setup is not ideal for Big Data analytics as it automatically puts several obvious constraints on what can practically be achieved. In *Chapter 3*, *Unleashing the Power of R from Within* we gave you a short introduction to parallel computing in R and we also mentioned that certain R functions can perform much better when more than one CPU/core is engaged. In this chapter we will take it a step or two further. We will not only be able to potentially increase the number of cores at our disposal, but we will also be in a position to expand the available RAM resources and optimize our cloud platform to easily analyze and manipulate Big Data using R. In this chapter, you will:

- Understand the concept of cloud computing for Big Data management and analytics
- Be introduced to major cloud computing providers (for example, Amazon Web Services, Microsoft Azure, Google Cloud Platform) and the services they offer
- Learn how to create user accounts and launch Linux virtual machines with fully-operational R and RStudio Server distributions on Amazon Web Services and Microsoft Azure.

Faster, bigger, cheaper – Big Data in the cloud

The last several years have probably seen the greatest IT revolution since the origins of the World Wide Web. It is likely that the exponential growth of tech start-ups and data-driven applications would never happened if cloud computing didn't exist. Data processing in the cloud has allowed numerous individuals and businesses to scale their workflows and applications and make them accessible to the public without large capital expenses and with a minimum of time and resources spent on their deployment. Operations and processes, which in the past required vast investments, collaborative engineering knowledge, and large server rooms with expensive multiple computing units connected through miles of cable, have suddenly become affordable and easy-to-maintain even for home-run businesses and individual researchers or students. Within a few years, a whole ecosystem of new Big Data tools, methods and approaches has emerged as a result of this revolution and R can now be very easily integrated with these tools to allow convenient and comprehensive statistical analysis and predictive modeling applied to larger-than-ever datasets.

Major cloud computing providers

Thanks to a growing number of cloud computing providers, you can benefit from several on-demand **Infrastructure-as-a-Service (IaaS)** or **Platform-as-a-Service (PaaS**) solutions. In this section we will present three major cloud platforms which you can use alongside R distributions to implement your Big Data analytics workflows or even develop and build your own cloud-based data-driven products (**Software-as-a-Service – SaaS**).

Amazon Web Services

Launched in 2006, **Amazon WebServices** (**AWS**) (`https://aws.amazon.com/`) is operated by Amazon.com, Inc. and offers a wide array of on-demand, subscription-based online solutions and large data applications without any need to build physical server rooms or farms. Developers can securely store and quickly access their data as well as build their own tools and apps and deploy them to end-users through a selection of readily available AWS products and services.

The most well-known tools accessible to data scientists and developers through AWS include:

- **Amazon Elastic Compute Cloud (EC2)**: Allows users to benefit from scalable commodity hardware through virtual machines (*instances*) which can be customized by users and may contain any desired software (including R or RStudio). The word *elastic* refers to EC2's convenient scalability and its on-demand billing (charged per hour).
- **Amazon Elastic MapReduce (EMR)**: Provides a cost-effective and fast platform for distributed Big Data processing and analysis. It contains a managed *Hadoop* framework which may operate on clusters of EC2 instances and can also feed the data from Amazon data buckets such as **Amazon Simple Storage Service** (**S3**) or NoSQL and SQL databases, for example, **Amazon DynamoDB** and **Amazon Relational Database Service** (**Amazon RDS**).
- **Amazon Lambda**: An implementation of the *Lambda Architecture* for a scalable, real-time data processing (charged by every 100 ms). It allows the execution of specific code in response to certain triggers or user actions and it supports other AWS services and data stores.
- **Amazon Kinesis**: Enables rapid processing and analytics of large amounts of streaming data in real-time. The aggregated data can then be further used to create custom applications.

All of the above tools integrate very well with data storage solutions offered by AWS. As previously mentioned, AWS users can store and retrieve any amount of data through the S3. The access permissions and data storage security policies can also be controlled with the **AWSIdentity and Access Management** (**IAM**) tool. The data can also be accessed through Amazon's connectors and alternatives to traditional SQL and more flexible NoSQL databases through Amazon RDS and Amazon DynamoDB, respectively. The former, relational SQL-based database management system, offers a fully-managed, simplified and scalable server for MySQL, Oracle, SQL Server, PostgreSQL and MariaDB databases. On the other hand, the Amazon DynamoDB service provides a fast and reliable distributed NoSQL database server with data stored by default on **solid-state disks** (**SSD**), and its pricing is dependent on throughput rather than requested storage capacity. DynamoDB comes with its own flexible schema and its data model is based on tables (which unlike in standard relational databases only require the primary key to be defined), items, and their attributes. Specific details on the data model supported by DynamoDB are available at: `http://docs.aws.amazon.com/amazondynamodb/latest/developerguide/DataModel.html`.

On top of these data analysis and storage solutions, AWS offers a highly-scalable (up to petabytes of data), hosted, fault-tolerant, and SQL-based data warehouse service in the cloud called **Amazon Redshift**. Its powerful data crunching capabilities rely on the **Massively Parallel Processing** (**MPP**) architecture that distributes and parallelizes SQL queries across high-performance, specialized, and flexible data warehouse nodes. The capacity of these nodes largely depends on individual needs and can be easily adjusted for specific data processing jobs. In general, there are two types of Amazon Redshift nodes. **Dense Storage** (**DS**) nodes are very cost-effective as they make use of cheap traditional **hard disk drives** (**HDD**) to store massive amounts of data. For more computationally demanding tasks, **Dense Compute** (**DC**) nodes provide users with high-performance architecture including **solid-state disks** (**SDD**), very fast CPUs, and greater memory resources. As Amazon Redshift is an on-demand service, the hourly pricing is dependent on the specific architecture chosen and provisioned by the user and the geographical region where the nodes are hosted. Please be aware that for huge amounts of data continuously being processed through Redshift, your bill may eventually reach thousands of US dollars per annum, however the pricing for one-off Big Data processing tasks starts from as little as $0.25 per hour.

Apart from the already presented services, AWS contains other interesting platforms related to Big Data management and analytics for example **Amazon Machine Learning**, which allows a fast deployment of machine learning and predictive modeling algorithms and the assessment of their quality through the built-in components supporting data transformations, model evaluation, and interpretation as well as data visualizations. All AWS that can potentially be of any interest to Big Data analysts are available at: `https://aws.amazon.com/big-data/`.

Note that each AWS service has its own pricing table displayed on its relevant website, so it is recommended that you always check the terms and conditions of use and details of exactly what costs your provisioned services are associated with. The prices may vary depending on the selected region of the used infrastructure, chosen architecture (for example, the type of instance), whether they are charged at an hourly rate (for example EC2 instances), storage capacity (S3), throughput (DynamoDB), or a combination and also other additional factors, for example in Amazon Machine Learning, users are charged an hourly rate and also based on the number of predictive models carried out for a specific application.

For new AWS account holders, Amazon offers free credits to use on small, *micro* type instances for a limited period of time. Although they won't let you crunch huge datasets and run sophisticated Big Data analytics, they will definitely give you a taste of cloud computing through AWS and may come in handy for testing your cloud-based data processing workflows on small subsets. Whenever possible we will be presenting methods that can be run on these micro-sized servers, but the majority of the techniques shown in the following sections of this chapter and in the subsequent chapters of this book should ideally be run on paid, larger instances for greater performance gains. Note that by following all the instructions contained within this book you may be charged by certain cloud computing providers. How much depends on your chosen configuration and other factors affecting the pricing.

Cloud computing solutions offered by Amazon have been used extensively by numerous businesses around the world for a variety of data management purposes. Some of the largest clients and the most famous AWS case studies include Netflix, Facebook, NASA, Pinterest, and many others. Services offered through AWS are known for their almost 100% fault-tolerance, automated backups, full support, constant health monitoring, very good compatibility, and high level of integration with other AWS and external services.

In the following section we will introduce you to a similar bundle of cloud computing services and tools offered by Microsoft.

Microsoft Azure

Known since early 2010 as Windows Azure, **Microsoft Azure** (`https://azure.microsoft.com/`) was re-branded in 2014 in order to compete globally for the top position in the IaaS market with the already fully-developed and widely used AWS. In fact, both vendors have dominated the commercial market of cloud-based IaaS solutions with **Google Cloud Platform** and **IBM Cloud** still lagging slightly behind the powerful duo.

Microsoft Azure offers similar on-demand, pay-as-you-go cloud services and products to AWS, which are predominantly focused on providing managed and integrated tools for analytics, high-performance computing, data storage, mobile and online applications, and other web solutions. From a data analyst's point of view, Azure presents an interesting alternative to AWS by offering the following cloud-based services:

- **Virtual Machines (VM)**: enable rapid deployment of scalable and fully-managed VMs with Windows Server or Linux (different flavors) operating systems with the usage billed on a per-minute basis. The cheapest and the most basic Linux VMs on Azure start from just $0.018 per hour and the high-performance, heavy-load optimized VMs (equipped with the latest generation CPUs, 16 cores, 112 GB of RAM and 800 GB SSD disk) may cost up to $1.50 per hour. Rates vary depending on geographical location and requested specific configuration of the VM.
- **HDInsight**: Microsoft Azure's alternative to the Amazon Elastic MapReduce service, allows heavy data crunching through a managed *Apache Hadoop* distribution which supports a large number of *Hadoop* ecosystem open-source projects, for example **Spark**, **Storm**, **HBase** (NoSQL database), and **Solr**, and also other data analytics tools like R and RStudio Server. The per-hour billing for the HDInsight subscription depends on several configuration factors such as the type of provisioned **HDInsight** clusters (for example Hadoop, HBase, Storm, Spark, and others), number of nodes in the cluster (usually 2 head nodes plus a varying number of data nodes, master/zookeeper nodes, worker nodes and many others depending on the type of clusters), the purpose of nodes (or the technical specification of instances), requested storage, and outbound data transfers, to mention just a few. The Azure pricing calculator available at `https://azure.microsoft.com/en-gb/pricing/calculator/` makes the budgeting for more complex services like HDInsight a little bit easier.
- **Stream Analytics**: A scalable, fault-tolerant platform for real-time analytics of streaming data enabling users to make sense of millions of events fed from Event Hubs Azure service every second. **Stream Analytics** is fully-managed and supports SQL-based data transformations.

In terms of data storage, Microsoft Azure provides several solutions ranging from standard relational databases known simply as a SQL Database, a NoSQL database system called **DocumentDB**, **Azure Blob Storage** for massive unstructured data files such as documents, movies, or other media files, **Azure Queue Storage** for messages, or **Azure Table Storage** for structured NoSQL data. All data storage services are highly scalable and provide users with almost 100% reliability, multiple backups, and optional geo-redundancy for enhanced data security and disaster recovery.

Apart from HDInsight and Stream Analytics, Microsoft Azure offers a number of other powerful analytical tools such as a cloud-based Machine Learning platform for predictive analytics, **Data Lake Analytics** (a multi-purpose and distributed Big Data management and processing suite), and **SQL Data Warehouse** (an elastic, enterprise-style data warehouse service with *massively parallel processing* and built-in support for Transact-SQL queries, allowing integration of non-relational data with structured, relational SQL databases). All Azure services complement one another pretty well enabling users, for instance, to attach, the chosen storage solutions to different Azure analytical and computing tools or integrate with other proprietary and open-source projects like Apache Hadoop or R.

As of February 2016, new users can enjoy a one-off $200 credit to spend on selected Azure services with additional free options and considerable discounts available during the trial period. Of course, this allowance may vary depending on your location, your specific circumstances and business needs, or they may simply change with time, so remember to check directly at the Microsoft Azure website (`https://azure.microsoft.com/`) what incentives are currently on offer. Another way of obtaining free or reduced Azure services is by joining the **BizSpark programme** for newly-formed businesses (see `https://azure.microsoft.com/en-gb/pricing/member-offers/bizspark-startups/` for more details). If your start-up meets certain eligibility criteria, for example it's been in operation for less than five years and hasn't reached a specific turnover, you and your company may be selected to enjoy free or discounted subscriptions to some or all Microsoft Azure tools and solutions for up to three years.

So far we have reviewed two leading providers of cloud computing services, which we will be using in the following chapters of this book. To complete the presentation of major IaaS vendors, we have to introduce you very briefly to a growing platform operated by tech giant-Google.

Google Cloud Platform

Google Cloud Platform is the youngest of the three major cloud computing providers presented in this chapter. Originated from **Google App Engine** (2008) and **Google Cloud Storage** (2010), it has only recently (2013) ripened into a more mature high-performance computing platform with the release of the **Google Compute Engine**-an alternative to Amazon EC2 instances and Microsoft Azure virtual machines.

Currently Google Cloud Platform contains several highly-scalable and fully-managed IaaS and PaaS solutions including Big Data storage, computing, and analytics tools such as:

- **Compute Engine**: A high-performance virtual machines, priced on a per-minute basis after the first 10 minutes; varied in CPU (up to 32), data storage, and memory capacity (up to 208 GB), with the ability of building custom machines
- **Cloud Storage**: Traditional object storage service known for its durability and scalability
- **Cloud Bigtable**: A massively scalable, high-performance Big Data NoSQL database, with integrated support for the Hadoop ecosystem including HBase and Spark, and compatible with other Google Cloud Platform services; priced separately for nodes (minimum three nodes) and data storage (SSD or HDD); Google search engine and Google end-user products like Gmail operate on a Bigtable database
- **Cloud Datastore**: A schema-less, fast NoSQL database optimized for web and mobile applications; comes with a free version with daily usage limits of stored data (1 GB) and number of read and write operations (50,000 for each type)
- **Cloud SQL**: A highly-scalable, improved and faster SQL-based relational MySQL database hosted and fully-managed on Google Cloud Platform
- **BigQuery**: A fast data warehouse for Big Data analytics allowing standard SQL querying and even real-time analysis of streaming data.

Other services offered by Google Cloud Platform which specialize in Big Data processing include: **Dataflow** (designed for batch processing and management of large amounts of streaming data), **Dataproc** (allowing Big Data processing through a fully-managed Hadoop MapReduce framework, Spark, Hive, and Pig), and **Datalab** (an interactive tool based on **IPython/Jupyter** that enables Exploratory Data Analysis with dynamic visualizations).

Similar to AWS and Microsoft Azure, Google Cloud Platform products can be flexibly arranged in a fully-compatible architecture as users mix and match specific services they need for their applications and data analysis tasks.

For new users Google offers $300 of credit to spend on user-chosen cloud services over a period of 60 days. Specific pricing policies for all Google Cloud Platform products are presented and explained at `https://cloud.google.com/pricing/`, where you can also find a link to a pricing calculator which may help you in estimating your overall cost. As previously stated, we strongly advise you to check the current charges before subscribing to any of the services and tools described in this book.

We will finish the first part of this chapter by reviewing general security and compliance policies related to data stored in the cloud by the presented vendors.

Data security and compliance in the cloud

Many users and organizations which deal with highly-sensitive or disclosure information may probably still have some doubts as to whether their data can be securely stored and administered on cloud servers without compromising the safety and privacy of their data, know-how, and business operations. The leaders in the cloud computing industry, such as Amazon, Microsoft, and Google, take this matter extremely seriously and privacy breaches or data leaks are now very uncommon on cloud services hosted by these organizations. If they ever occurred, such events would be a massive blow to the reputation of these businesses and have a disastrous effect on the market values of the involved companies, as well as possibly threatening their very existence. The Sony data breach in 2014 and TalkTalk cyber-attack in November 2015 are the most recent examples of data security fiascos on a huge scale that shook these businesses very badly and left multi-million dollar scars on their annual financial statements.

The industry standard, and the absolute must-have for any respected cloud vendor is that the service has to comply with stringent, third-party, independently-audited data **security certifications** such as **ISO 27001**, **SSAE16**, and ISAE 3402 Type II standards (for example **SOC 2/3**), and **PCI DSS 3.0**, as well as certain international data protection rules and guidelines such as the **EU Data Protection Directive**. These certifications and audits ensure that not only individual or consumer data and rights are protected but, more importantly, services, technology, systems, tools, applications, and employees involved in the provision of cloud products comply with the highest security and privacy standards. Besides, certain companies may adopt additional safety and privacy measures, which are often self-imposed and only internally-audited, but sometimes may equally be imposed by local or national governments depending on the specific geographical location of their business activities. These additional data security and privacy regimens may for instance relate to setting custom permissions and access credentials to data stored by users in the cloud, specific methods of data disposal (**digital shredding**), authorization layers for logging in to provisioned services, networking issues and network intrusion detection, data encryption standards, security and anti-virus scanning, country-specific vetting procedures for engineers with unrestricted access to customer or user data (for example a former **Criminal Record Bureau** certificate, now re-branded as **Disclosure and Barring Service**, check in the United Kingdom), and many others.

A vast majority of the leading cloud computing vendors provide clear statements on their current and up-to-date data security policies and obtained compliance certificates. It's also no secret that many financial institutions (for example Bloomberg), international enterprises (Facebook, Twitter, and many others), and even national authorities (governmental agencies like, CIA) or political organizations use cloud computing on a daily basis. If they can accept the risks and threats of using cloud services, then individual users, researchers, and small- or medium-sized businesses can probably do so too.

RStudio Server in the cloud

In the preceding section we have presented a variety of tools and products including free and trial offerings provided by Amazon, Microsoft, and Google Cloud solutions. In this section we will get our hands dirty and will guide you through the installation process of the core R and RStudio Server distributions on Linux virtual machines available through Amazon EC2 and Microsoft Azure. The tutorials will include instructions on how to:

- Create cloud computing accounts and deploy the most basic virtual machines (they are free if you are a new user)
- Install and configure core R and RStudio Server and its dependencies
- Use the command line (through a shell/terminal) to set up and configure created virtual machines

RStudio Server on Amazon EC2 instances

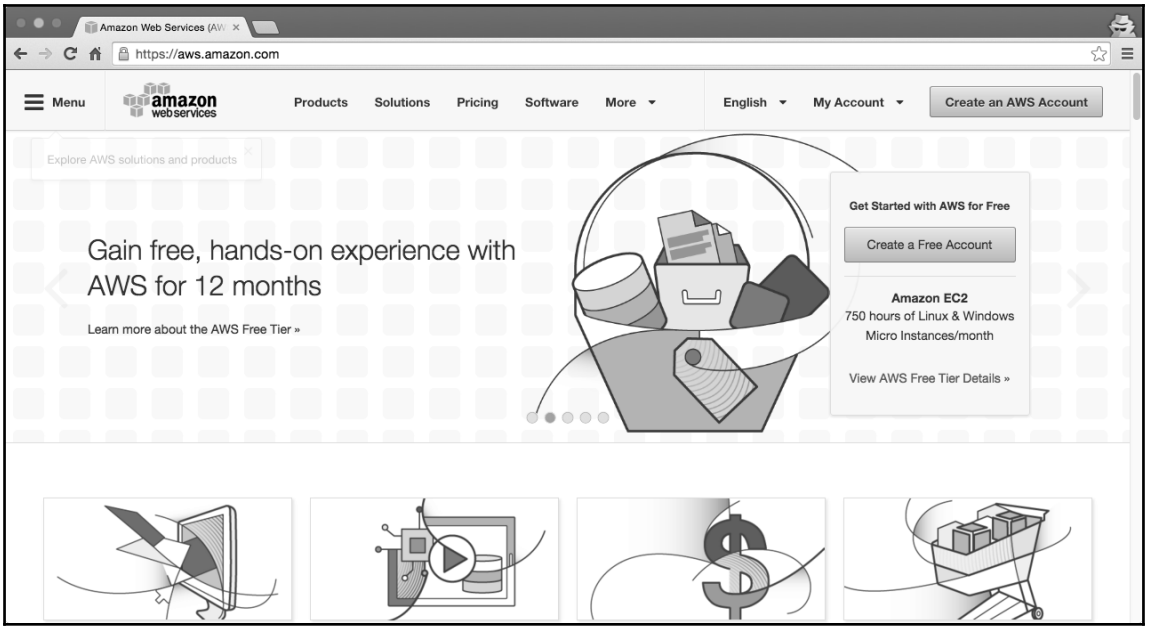
We will begin a practical tutorial for creating an AWS account for a new user.

Please note that the instructions and services described in this guide relate only and explicitly to **AWS Cloud computing services** available at the end of January 2016. It is likely that the instructions and guidelines provided in this section may differ widely from the actual AWS registration procedures after the end of January 2016, therefore please apply them with care, especially when subscribing to paid services.

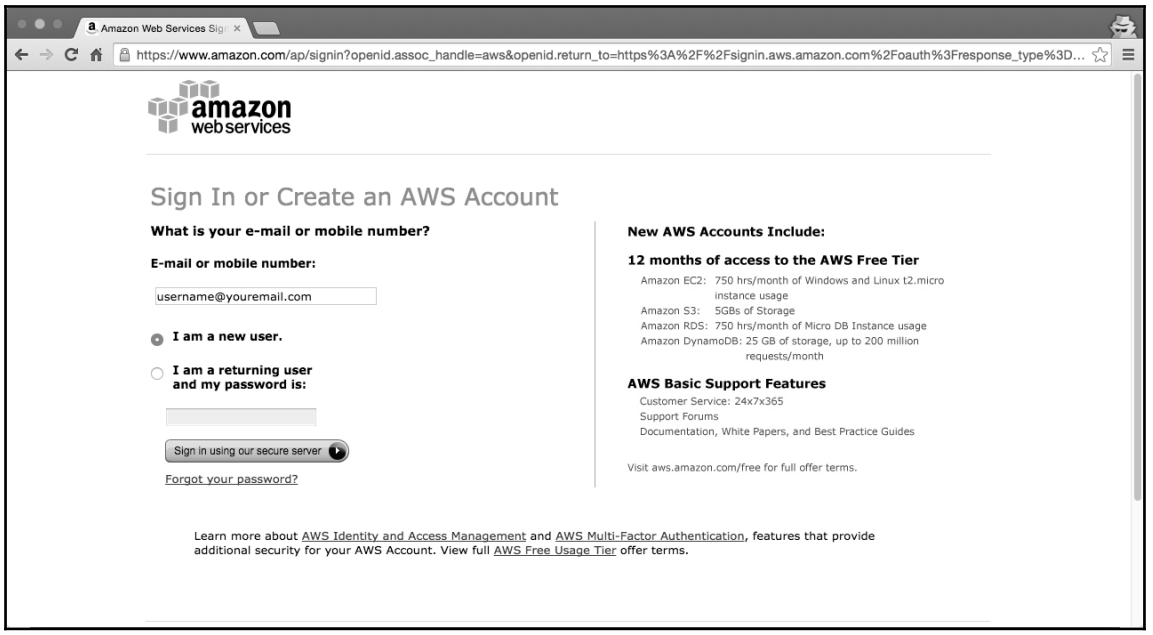
Creating an Amazon Web Services user account

In order to benefit from the 12-month *Free Usage Tier* AWS offering, prospective users should create a new account and verify it by providing personal details such as current address, email address, a valid telephone number, and credit/debit card details. As long as you do not go beyond the set limits and avoid using more powerful infrastructure than the one specified in the free tier terms and conditions you will be able to use selected services free-of-charge. Again, please check what services are included in the Free Usage Tier membership at the time of your registration. The following procedures will guide you through the process of creating a new account on AWS:

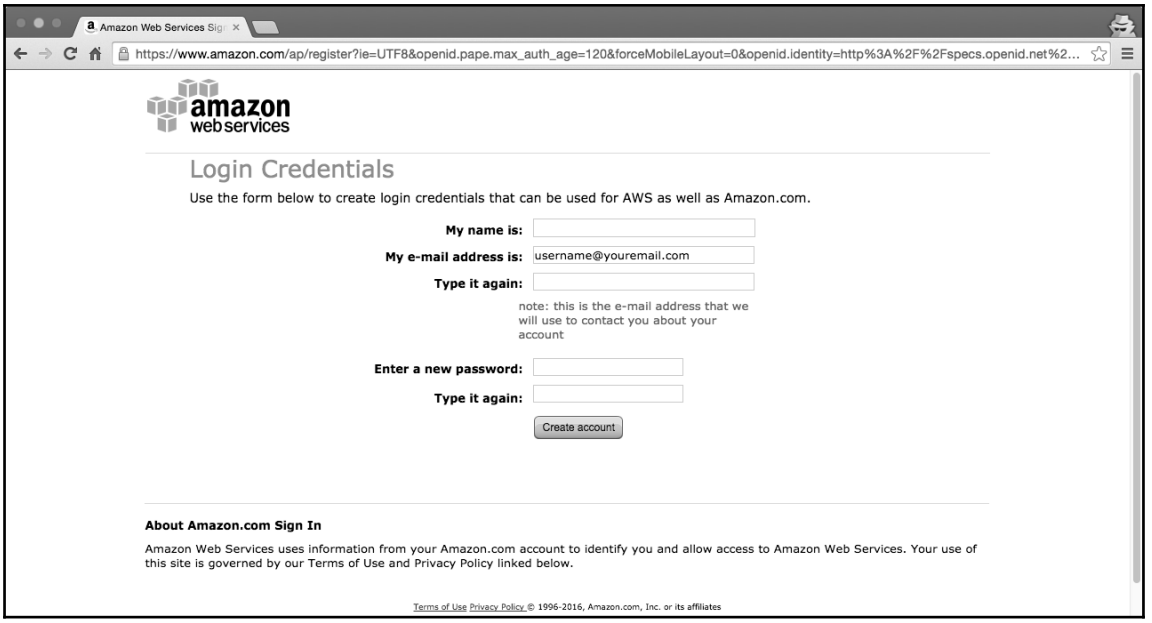
1. Go to `https://aws.amazon.com` and click on the **Create a Free Account** button. You will be taken to a new page that will let you sign in to an existing account or create a new user account. It will also list a number of services included in the Free Tier subscription:



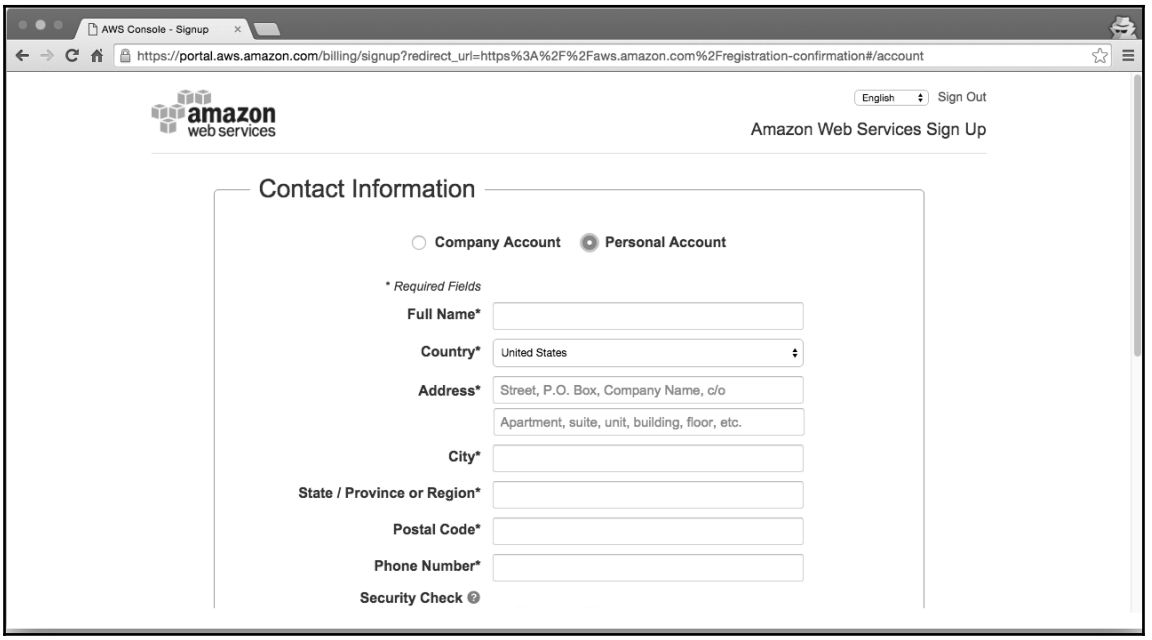
2. Type your email address (or mobile number) and select the **I am a new user.** option. Click on the **Sign in using our secure server** button to continue:



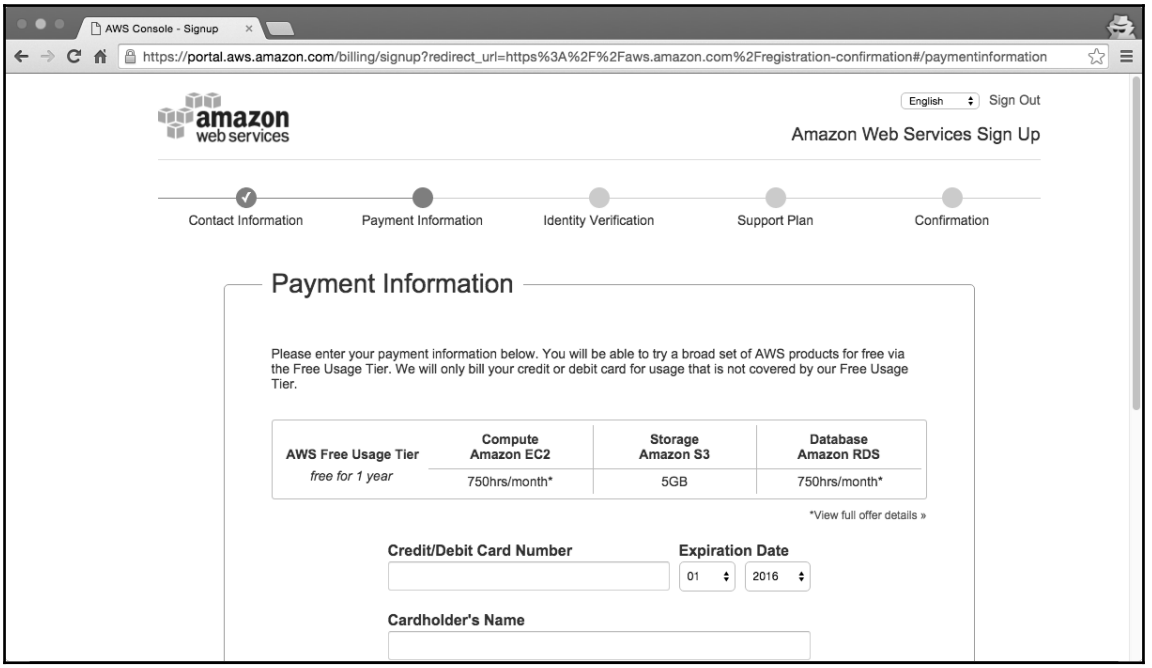
3. Complete the form to create login credentials (name, email, and password) and press **Create account** to proceed:



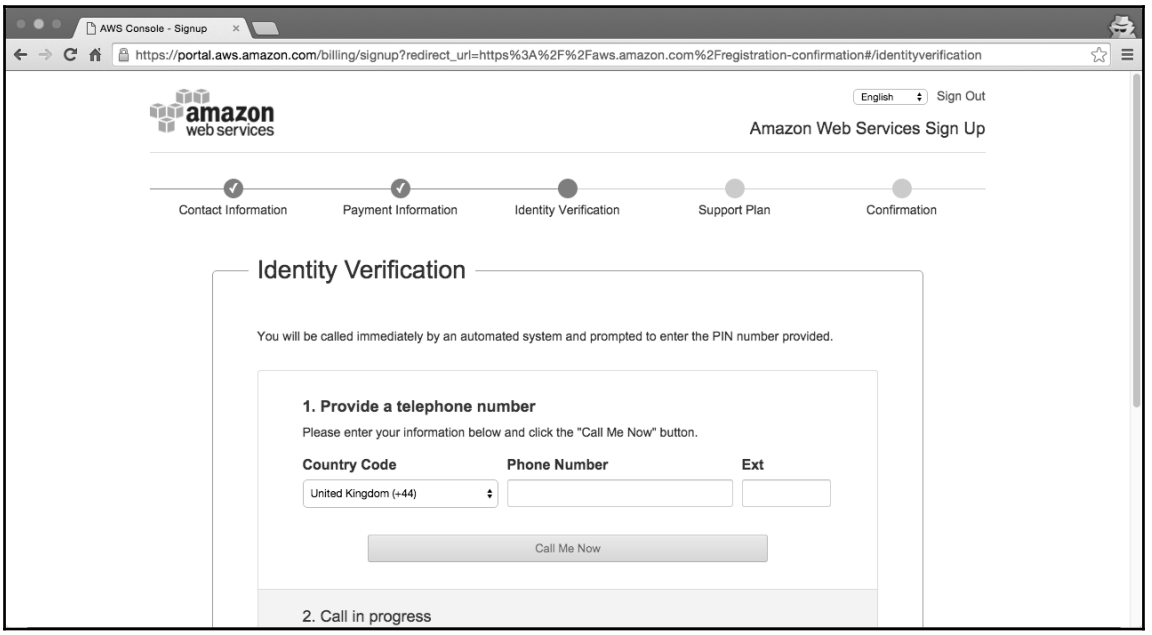
4. Regardless of whether you sign up for a personal or company account, you have to provide valid contact information:



5. In order to proceed you need to complete the payment information form by providing your debit or credit card details. As a new user registering for the 12-month Free Usage Tier, you won't be charged anything unless you use services and products that are not covered by the Free Usage Tier offer. The following screenshot lists three AWS services included in the Free Usage Tier, however their usage is capped by further restrictions applied to the storage capacity of Amazon S3, type of instance used, and other limitations. Make sure that you always read and understand details of specific offers that you subscribe to through Amazon or other cloud computing providers:

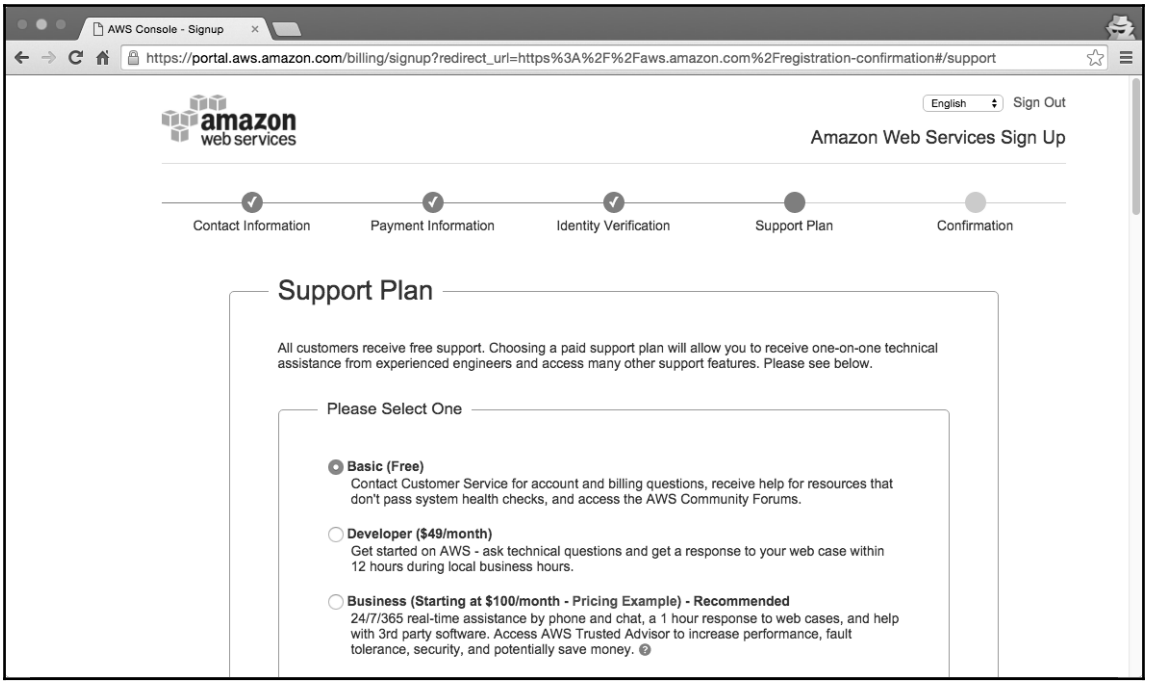


6. Your identity will now be verified. You need to provide a direct phone number and press the **Call Me Now** button to receive an automated phone call from Amazon. The screen will reveal a four-digit PIN number which you will have to either say or type by pressing your telephone keypad:

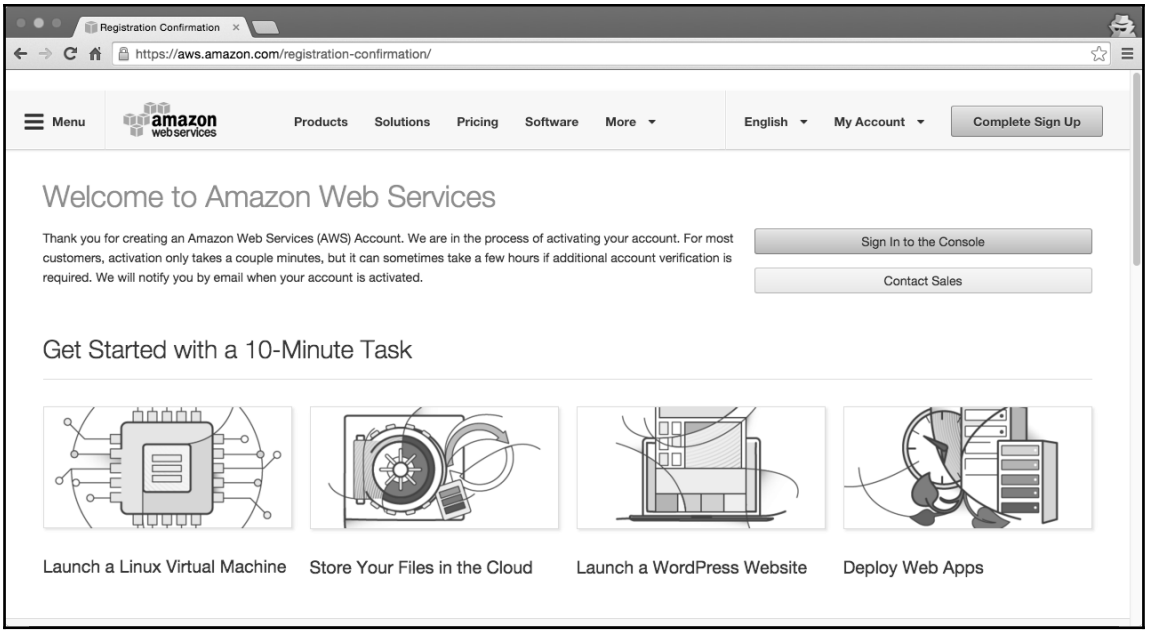


Upon successful verification of your identity, you may press **Continue to select your Support Plan** to choose one of the help and support plans on offer.

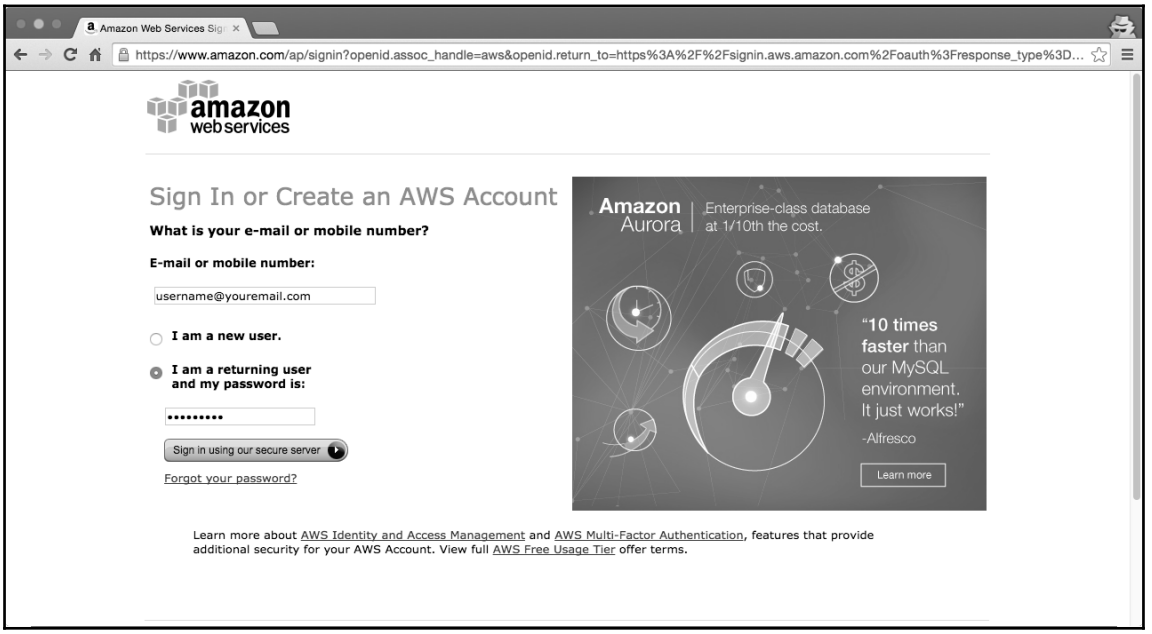
7. There are four different Support Plans available to AWS users: **Basic (Free)**, **Developer ($49/month)**, **Business,** and **Enterprise**. The Basic plan is free (included in the Free Usage Tier) and at this stage it should suffice. Once your computing needs and required infrastructure expand you may wish to upgrade it to the Developer level or higher depending on your specific circumstances. If you scroll down the page you will find more details and comparisons of all Support Plans available to AWS users:



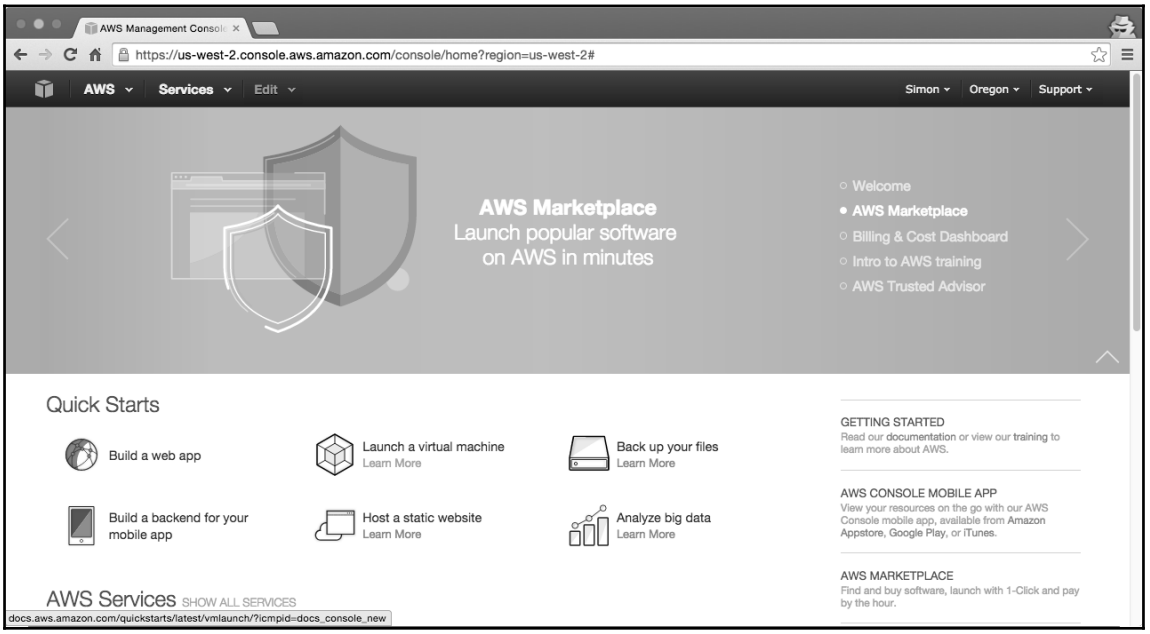
8. The selection of a Support Plan completes the registration process. You will be redirected to the registration confirmation page and should have now received a couple of e-mail messages at your designated e-mail address. The activation of your AWS subscription may take anything from a few minutes to several hours depending on whether there is any additional information that must be verified. You will get a final registration confirmation or further registration-related communication by e-mail:



9. Once your registration is complete and your account activated, you may now log-in to the AWS console by providing your login credentials, which you set up in step 3:



10. Upon logging in you should be able to see the front page of the console with the upper navigation bar and the shortcut icons in the main body of the page, similar to the one presented below:

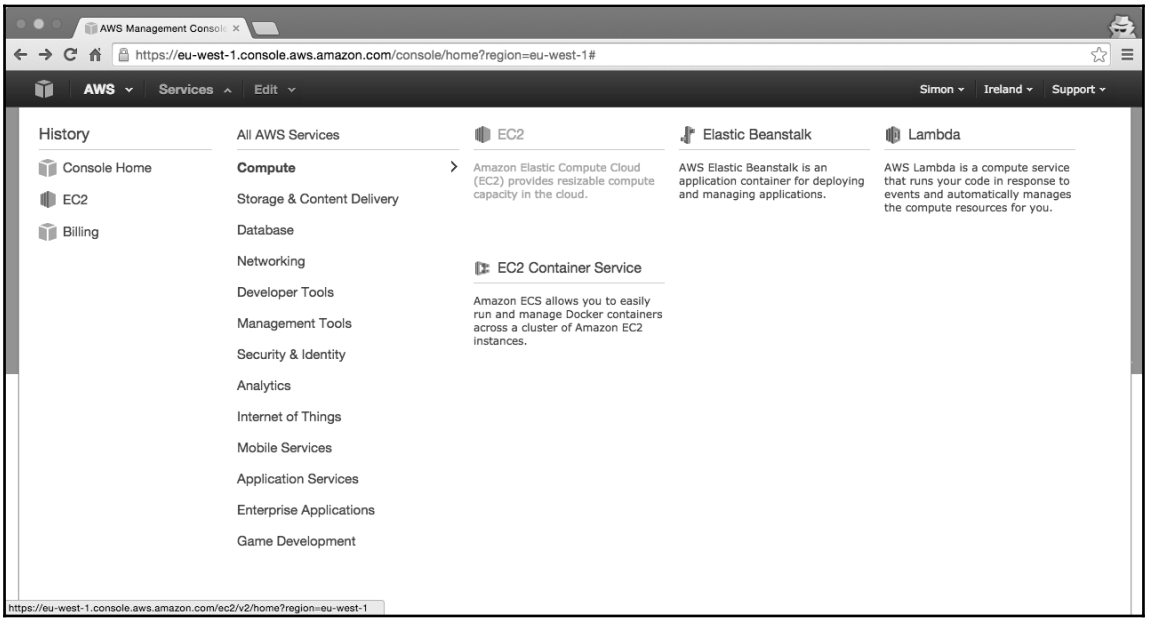


Congratulations, you have now become a new AWS user. In the next section you will create your first basic EC2 instance.

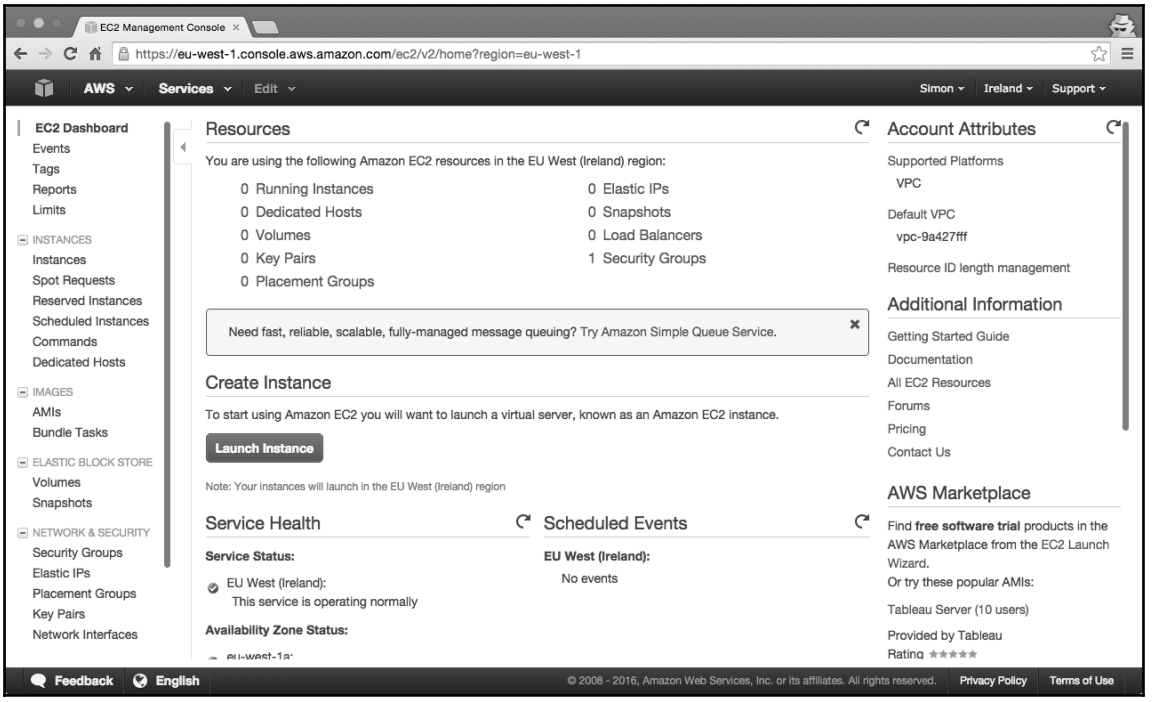
Creating your first Amazon EC2 instance

In this section we will create a free *t2.micro* EC2 instance. It is currently the only EC2 instance that is included in the Free Usage Tier and, according to the current offering (as of the end of January 2016), you can use the micro instance up to 750 hours per month, totally free. Obviously it's not the most powerful virtual machine users can choose, it only contains one virtual CPU, as little as 1 GB of RAM, and a low to moderate level of network data transfer. Although this is comparable to a starter, low-end, commercial laptop, users can at least test AWS services before purchasing much more powerful cloud solutions. The following instructions will guide you through the process of deploying your first EC2 *t2.micro* instance:

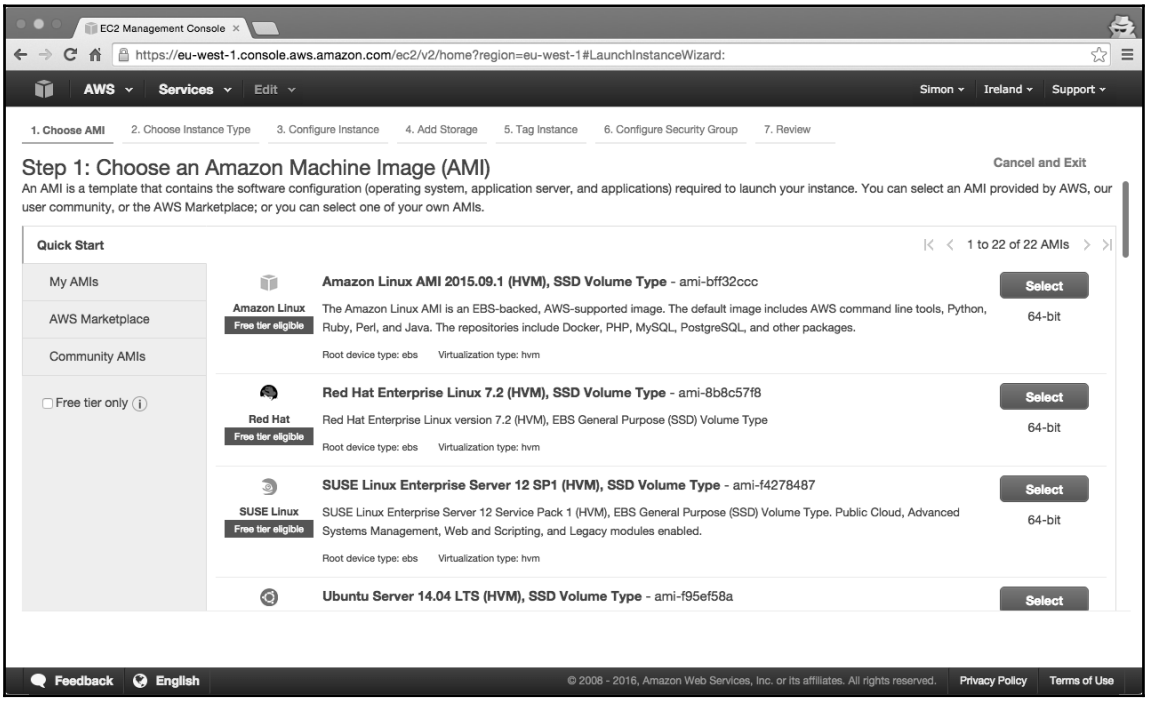
1. After logging-in to the console, click the **Services** tab in the top menu. When the whole list of available services unfolds, choose the **Compute** option and select **EC2** services as shown in the following screenshot:



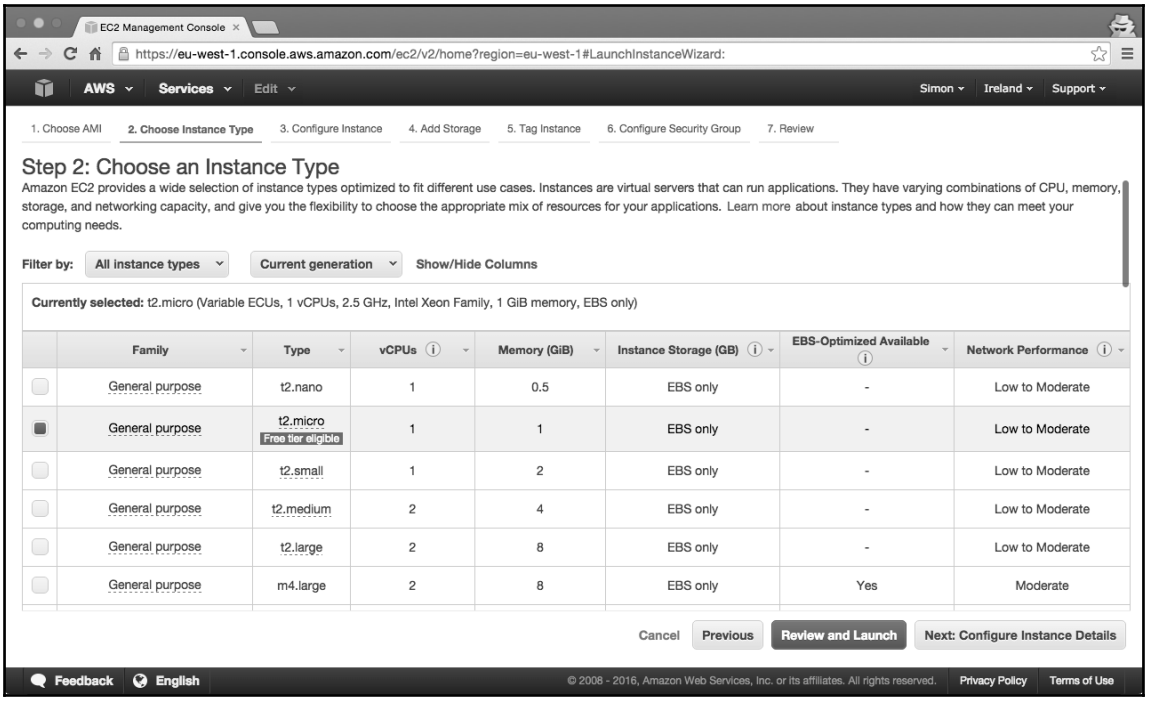
2. You will now be re-directed to the main **EC2 Dashboard**. It consists of: a right sidebar panel with shortcuts to various features and configuration options of EC2 instances; the main EC2 panel with information about deployed **Resources**, a shortcut to launch a new instance, and a **Service Health** console indicating any issues with the AWS servers located in your chosen region; a left sidebar panel with **Account Attributes**, shortcuts to help and support files and documentation (**Additional Information**), and finally links (**AWS Marketplace**) to fully-managed and ready-made trial or billed Amazon Machine Images (**AMI**) of proprietary Big Data tools such as Tableau Server or TIBCO Spotfire Analytics Platform. Choose **Launch Instance** from the main panel to start creating a new EC2 virtual machine:



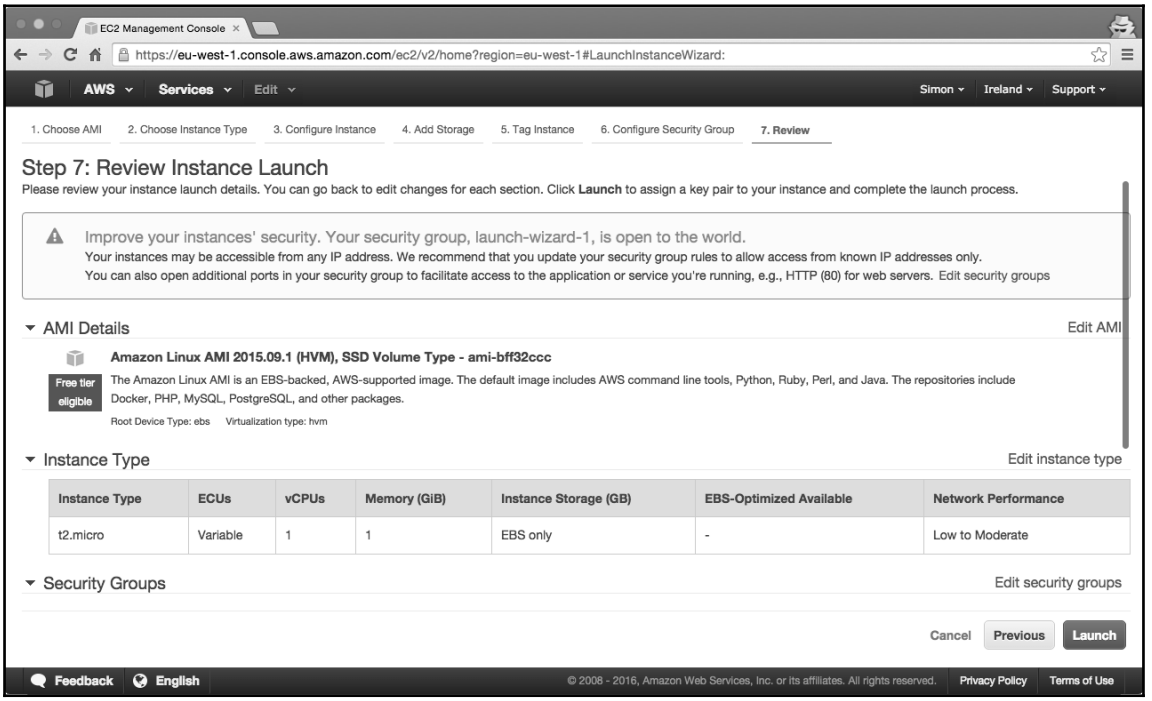
3. We will create a basic Amazon Linux AMI instance, which by default includes several packages that support command line scripting, Python, Perl, Java, MySQL, and several others listed in the instance's description. It's the most *no-fuss* VM-a perfect choice for an AWS newbie, however if you wish you can choose other Linux or Windows virtual servers that are included in the Free Usage Tier (**Free tier eligible**):



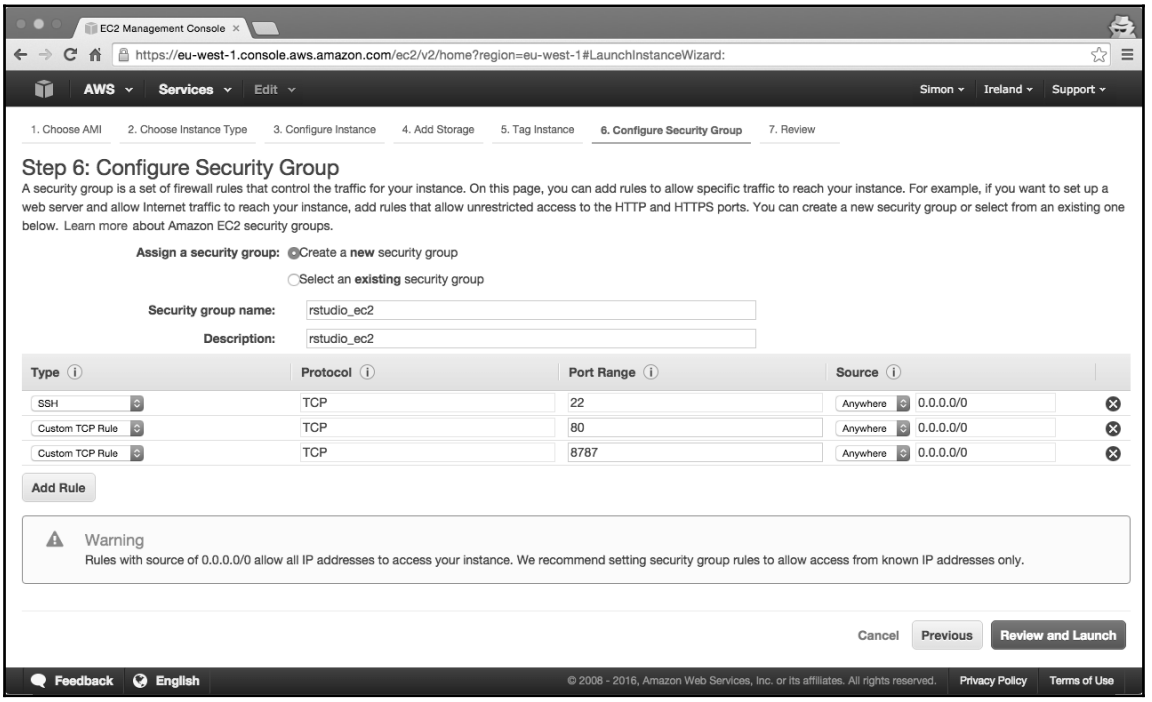
4. Once you decide on the OS of your virtual machine, you will proceed to the selection of the specific instance configuration. If you don't want to be charged, your choice will be limited to just one option – **t2.micro** instance. Let's then select it and click the **Review and Launch** button. At this stage we will skip more detailed configuration of our instance and we will also stick to the default EBS storage – we won't be attaching any additional storage systems to the virtual machine at the moment:



5. You will be taken directly to the last stage of the instance deployment. But our work won't end just yet. The review panel displays all features of our created (but not yet launched) virtual machine. Before its final deployment though we need to edit the settings of the firewall rules assigned to this instance to include ports that will allow internet traffic to reach our RStudio Server. In order to do this we must click on the **Edit security group** link:

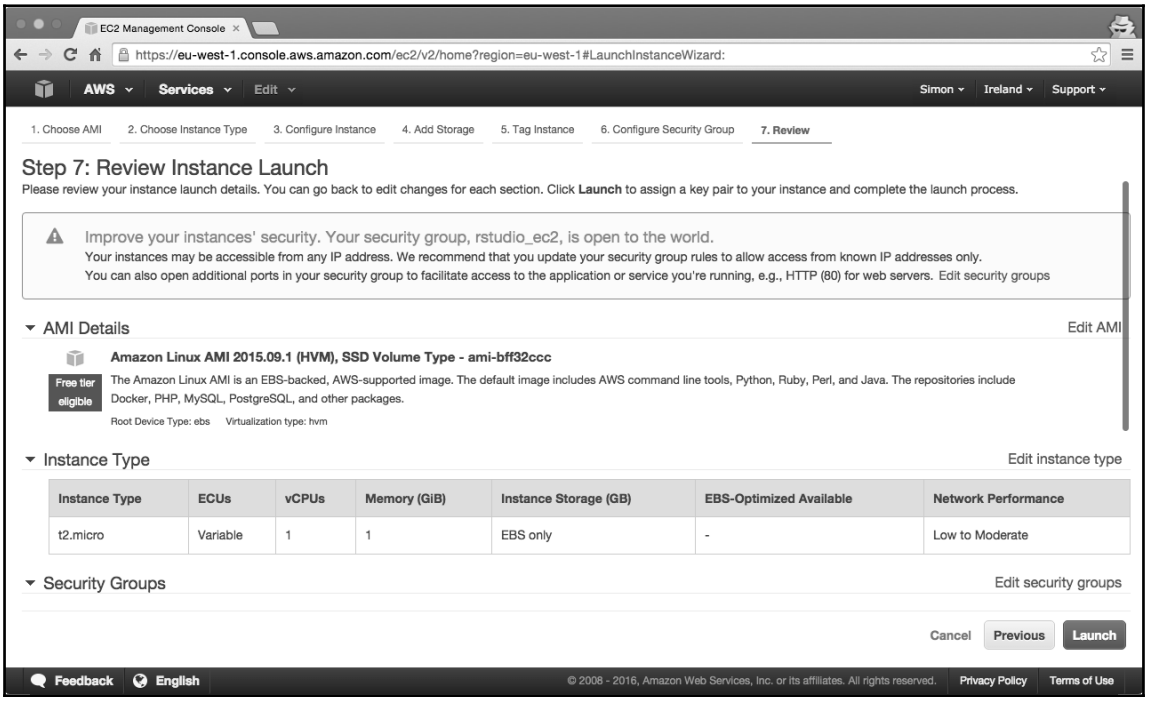


6. This will take us a step backward to configure our security group. Firstly, we have to create a new security group which we can simply call **rstudio_ec2**. In the description you may specify the name of the group again and add more specific information such as the date of its creation and other details. We will keep it very basic by simply repeating the name of the group. In order to add new firewall rules that allow unrestricted access to our RStudio Server, click the **Add Rule** button underneath the table with the existing rules and add two new custom TCP rules on ports `80` and `8787`. For the time being allow all IPs to access the instance by selecting the **Anywhere** option from the **Source** column drop-down menu. You should end up with the Security Group configuration as presented in the following screenshot:

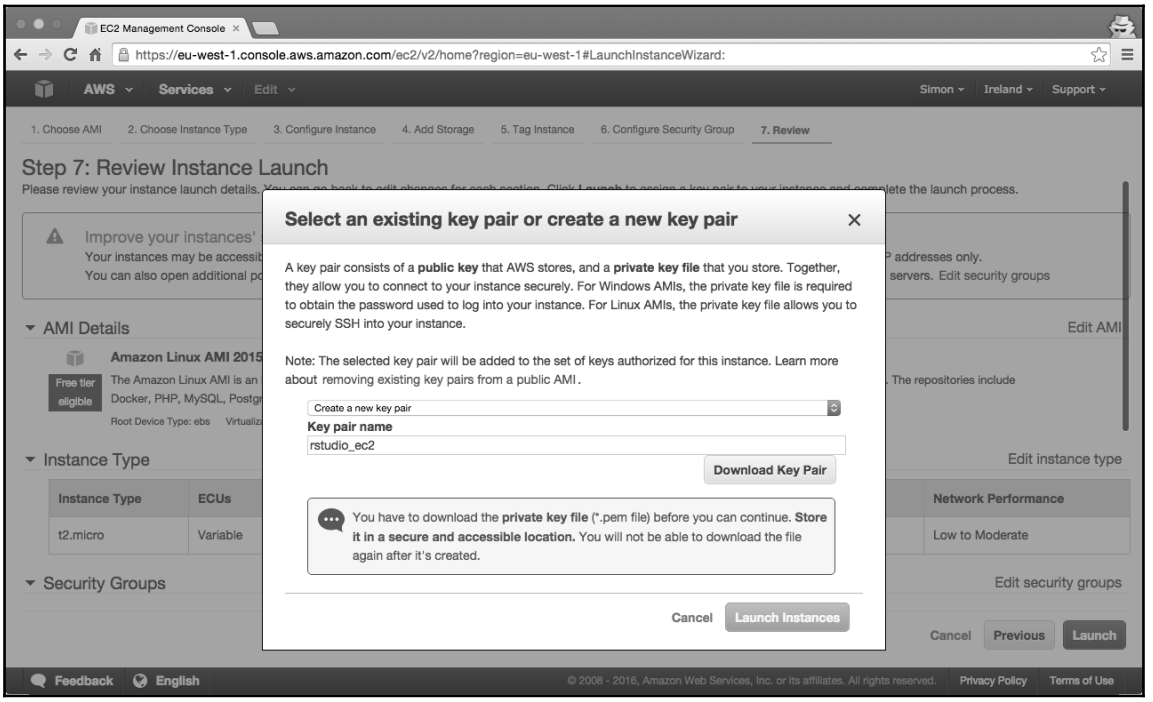


Press the **Review and Launch** button to accept the changes and go back to the Review page.

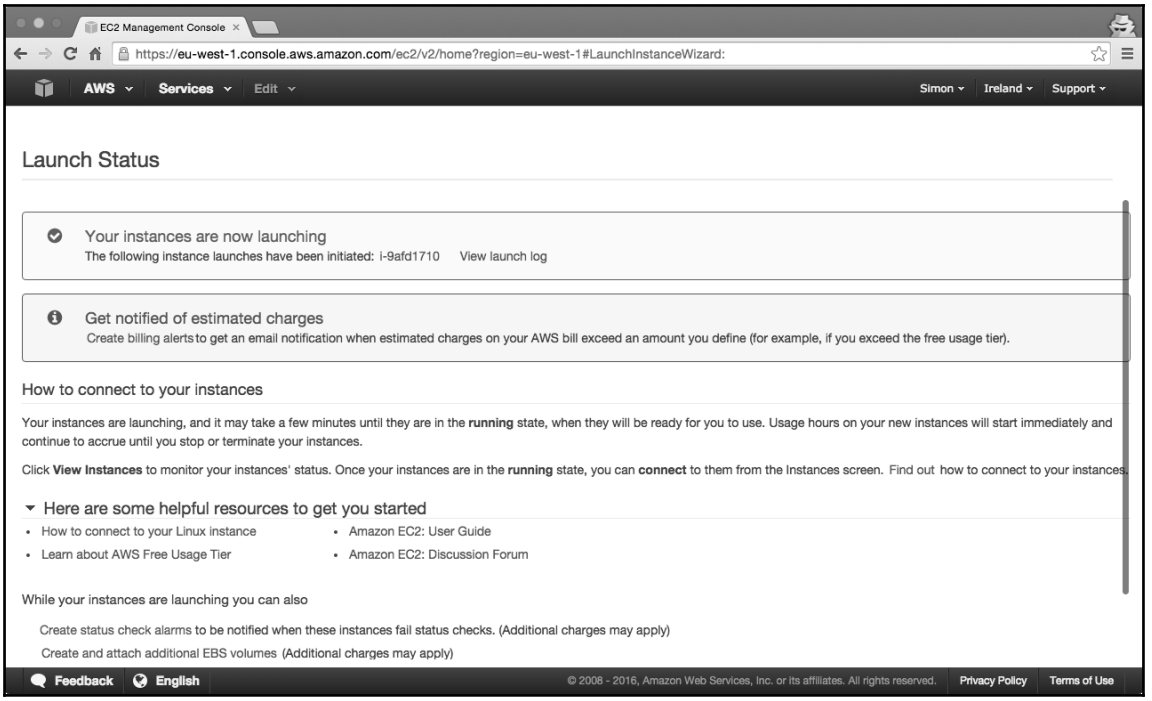
7. In the Review panel click the **Launch** button:



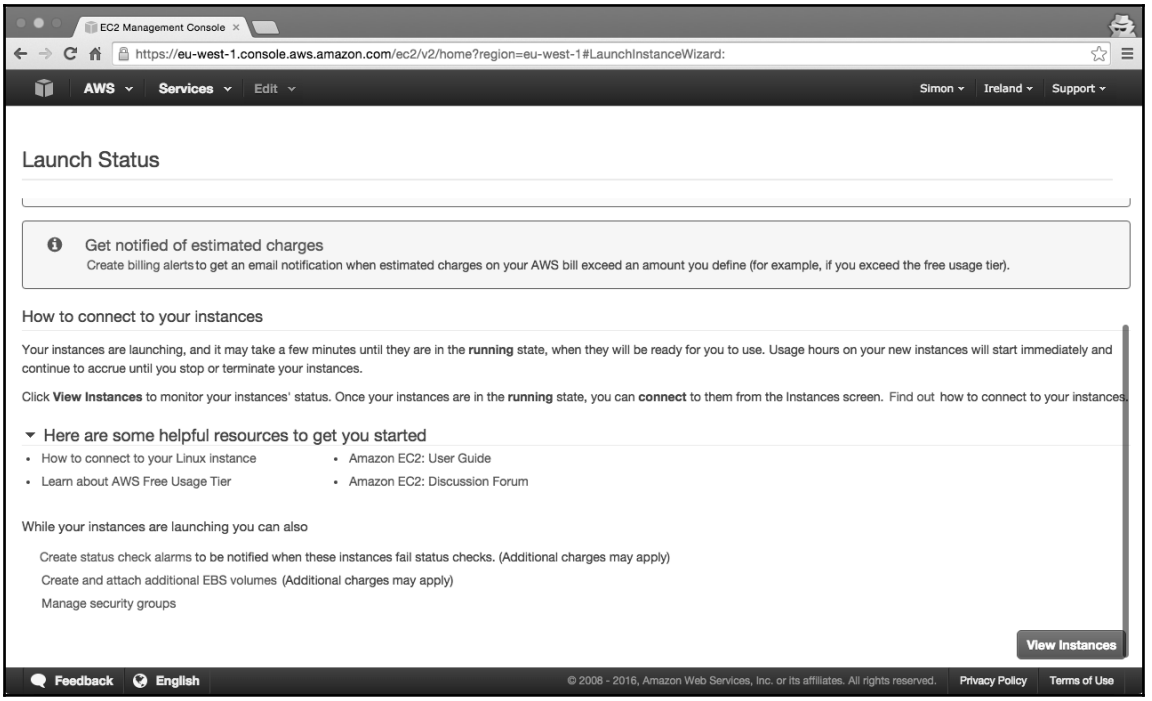
8. Launching the instance will activate a pop-up window which enables users to create and save a **key pair** (**a pair of public and private keys**) for Linux instances which in turn allows you to SSH to the virtual machine. You can name your key pair as you wish, but for simplicity we will name our instance the same as its security group, so `rstudio_ec2`. Proceed by clicking the **Download Key Pair** button. The `rstudio_ec2.pem` file will be downloaded to your default location for Internet downloads, but make sure to keep the file in a safe and accessible directory on your computer – you will need to specify its location in the following steps. Click the **Launch Instances** button when the download is finished to initialize deployment of the virtual machine:



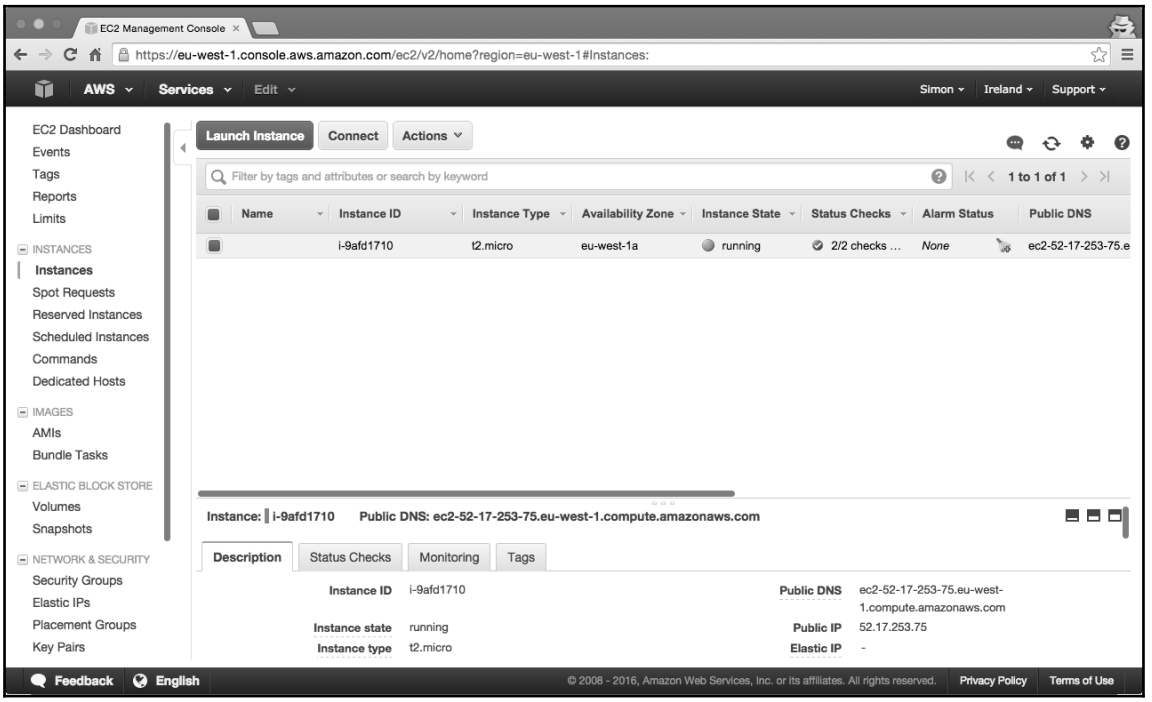
9. While launching the instance, you will be re-directed to the **Launch Status** page where you can inspect the job progress by clicking the **View launch log** link:



10. Scroll down to the very bottom of the **Launch Status** page and click the **View Instances** button:

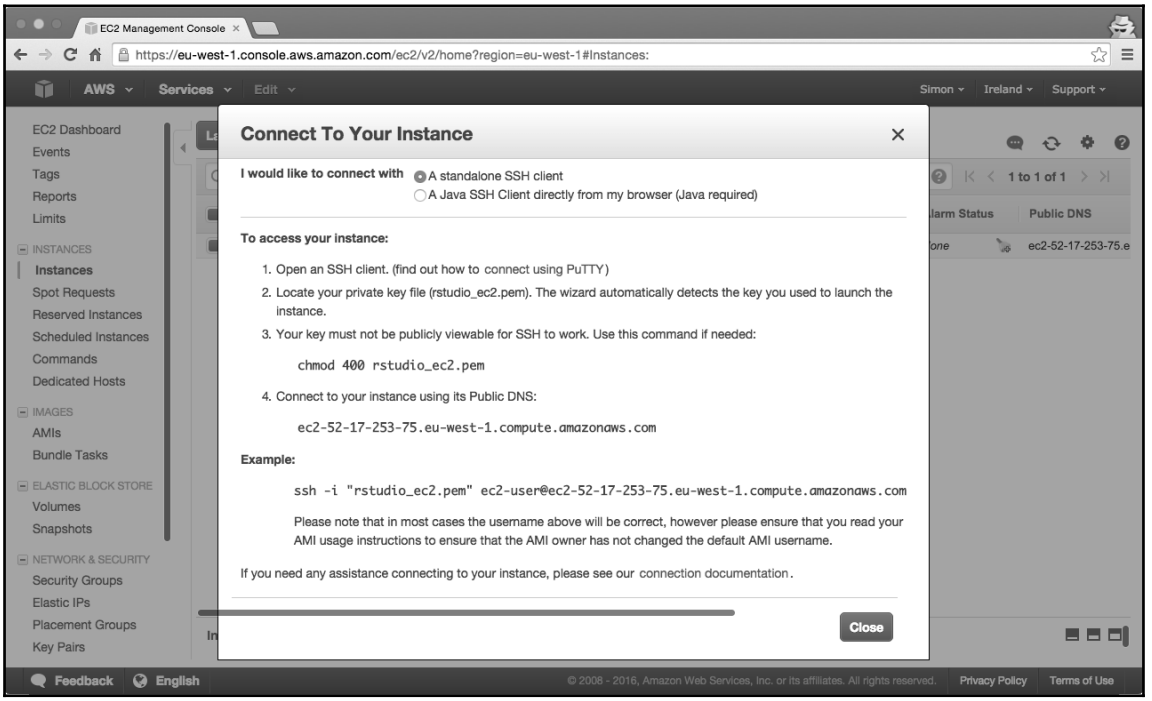


11. This will take you to a dashboard with all the created EC2 instances on your account. As you are a new user you should only see one virtual machine available in the main panel. When selected, you can inspect all important details about your instance (**Description** tab) such as its ID, public DNS and IP number, security group, key pair name, launch date, and many others:



There is also other vital information about the instance and its health provided by the **Status Check**, **Monitoring,** and **Tags** tabs.

12. Once the instance is launched and running, you may now update packages included in the AMI. To do so you first need to select the instance you want to connect to (if there is only one it's very likely it will be pre-selected by default). Then click the **Connect** button above. It will activate a new window (please see the following screenshot) with instructions on how to access the instance from a standalone SSH client like a shell or Terminal. If you are a Windows user, you can also SSH to the instance using the **PuTTy** tool (please click the **connect using PuTTy** link for more details). Alternatively, you may wish to connect through the browser-based Java SSH client:



13. At this stage, start your preferred SSH client and move to the location where you stored the `rstudio_ec2.pem` private key. For example, I've downloaded it to my `Downloads` folder, so I will simply use the following command to navigate:

```
cd Downloads/
```

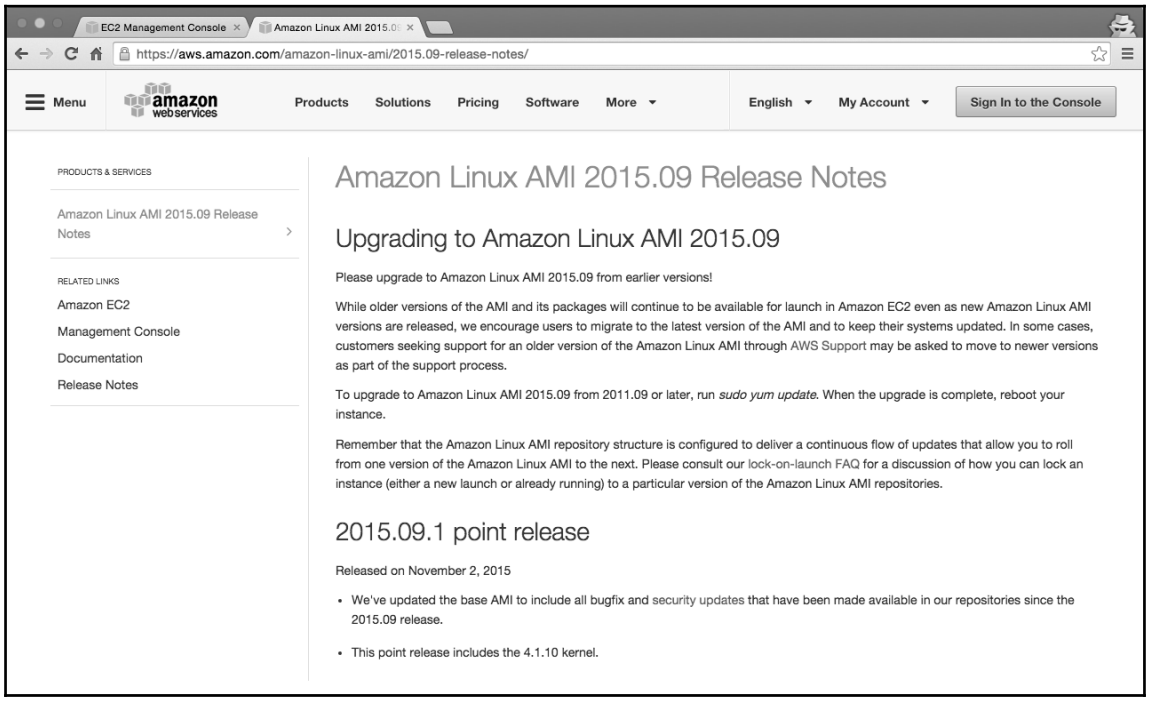
Make the downloaded private key inaccessible to the public with the following line:

```
sudo chmod 400 rstudio_ec2.pem
```

You can now SSH to your instance. For the virtual machine launched in the preceding steps, the command will be as follows (you will need to change the IP address to the IP of your instance):

```
ssh -i "rstudio_ec2.pem" ec2-user@ec2-52-18-27-20.eu-west-
1.compute.amazonaws.com
```

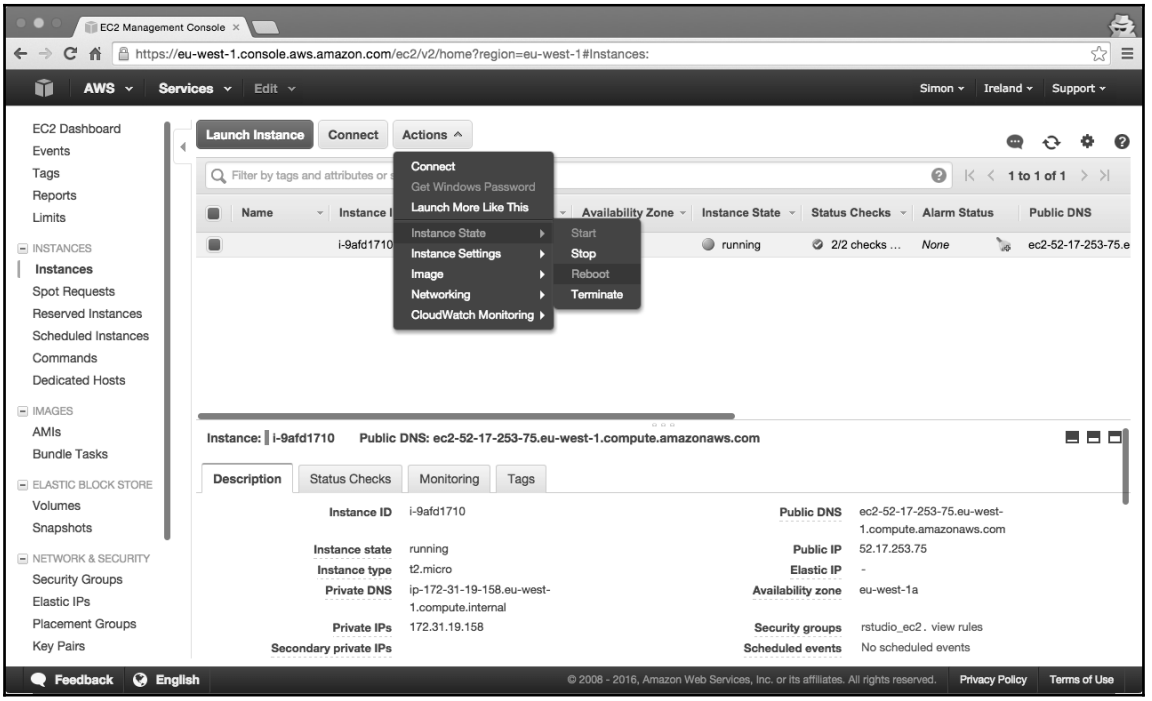
You should now be able to see the login confirmation output. It also includes a link to the release notes which describe details of the Amazon Linux AMI that we used for our virtual machine. In our case we used the Amazon Linux AMI 2015.09 version and its release notes are available at: `https://aws.amazon.com/amazon-linux-ami/2015.09-release-notes/`:



14. In case there are newer versions of packages installed in our 2015.09 AMI, we can update them with the following line:

```
sudo yum update
```

    This operation may take anything from a few seconds to several minutes depending on how many pre-installed packages have to be updated. Following the update make sure to reboot the instance by clicking the **Actions** button and selecting **Instance State** and **Reboot** from the drop-down menu as shown in the following screenshot:



    Confirm the reboot by clicking the **Yes, Reboot** button in the pop-up window.

This essentially completes the launch of the instance. In the following section we will explain how to install RStudio Server on our EC2 virtual machine.

Installing RStudio Server on an EC2 instance

As our instance is based on Linux 64-bit OS, we will be installing a 64-bit version of RStudio Server for Linux-operated servers. More specifically, our instance is running Linux CentOS Version 6.8, which is above the CentOS version 5.4 required by the current RStudio Server v0.99, therefore we do not have to enable the **Extra Packages for Enterprise Linux** (**EPEL**) repository. If for some reason your CentOS version is below 5.4, you can find the EPEL installation instructions at the following addresses: `https://fedoraproject.org/wiki/EPEL` and `https://fedoraproject.org/wiki/EPEL#How_can_I_use_these_extra_packages.3F`.

1. Apart from the above, RStudio Server requires that core R, available from the CRAN website is installed on the server. R version 3.2.2 should already be installed by default on the Amazon Linux AMI 2015.09 release instance, but if it's not there it can be easily installed using the following shell/terminal command:

```
sudo yum install R
```

2. The following is the base R installation you can now install for the current version of RStudio Server for Linux CentOS:

```
wget
https://download2.rstudio.org/rstudio-server-rhel-0.99.879-x86_64.rpm
...
sudo yum install --nogpgcheck rstudio-server-rhel-0.99.879-x86_64.rpm
```

   You may have to alter the version of RStudio Server by checking its current release at: `https://www.rstudio.com/products/rstudio/download-server/`. As of the beginning of February 2016, the most current version of RStudio Server was 0.99.879 and this release is used in the installation code above. The installation of RStudio Server may take a couple of minutes.

3. Once RStudio Server is installed, you may add users to the instance and specify their passwords:

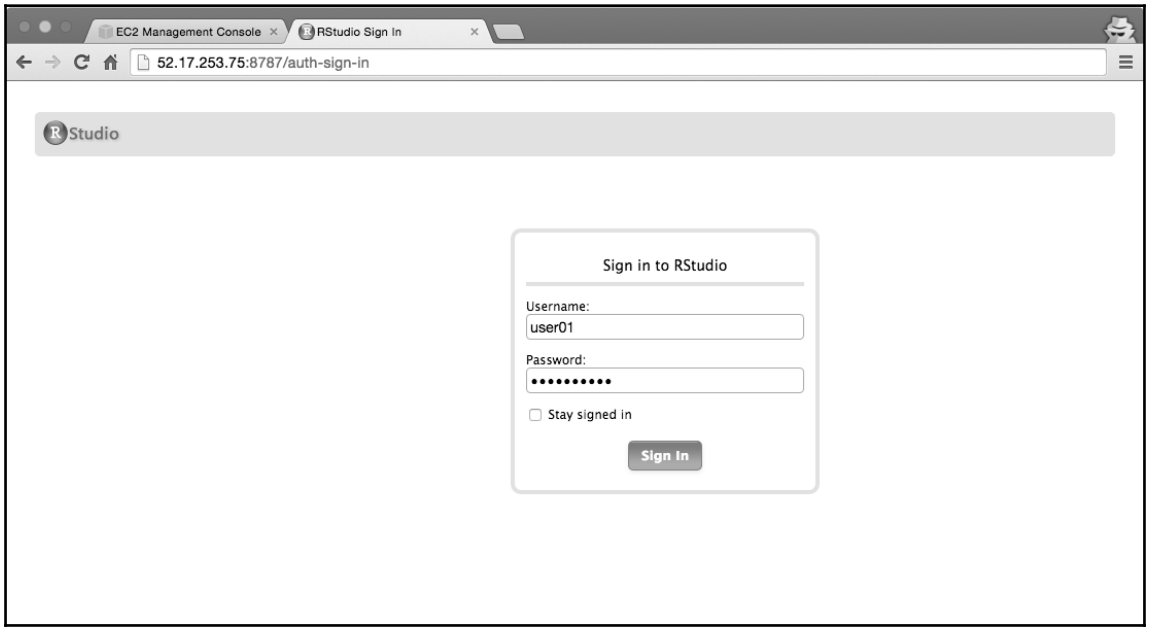
```
sudo adduser user01
...
sudo passwd user01
```

Here you will be asked to provide a new password for **user01** and confirm it. You may add as many users as you wish.

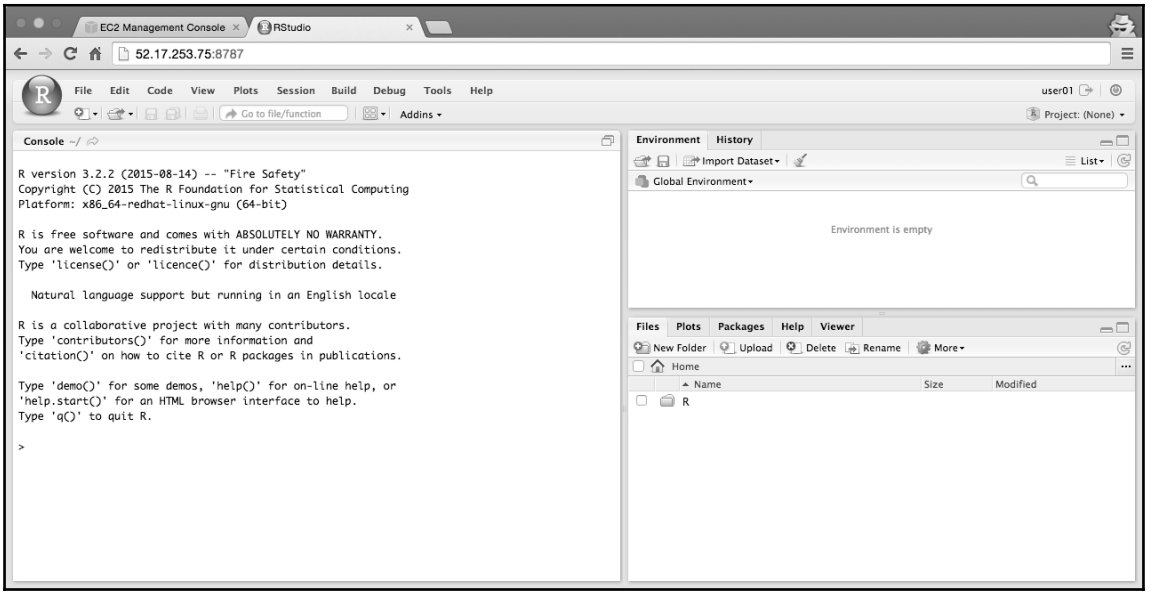
When you are done, you can just exit your shell/terminal:

```
logout
```

4. We can now check whether the installation was successful by logging-in to RStudio Server on the EC2 instance using our credentials for each user specified in the previous step. In order to do so, you need to point your browser (Chrome, Mozilla Firefox, or Safari are recommended) to the IP address of your running instance and the port which uses RStudio Server (`8787`). In our case the URL will be `http://52.17.253.75:8787`, but your IP will definitely be different so make sure to change it to the IP of your own virtual machine. If everything goes fine you should see the following RStudio Server sign-in screen:



5. Once logged-in, you should be able to see the RStudio Server application with all its panels as shown in the following screenshot:



RStudio Server is now fully installed and ready for use on your EC2 instance.

In order to avoid any additional charges in the future (beyond the Free Usage Tier) it is a good practice to *stop* and *terminate* the instance. Termination of the instance will also delete all existing files on your virtual machine.

Amazon Linux AMI – use it or not?

So far we've shown you how to launch and configure an EC2 virtual machine with the Amazon Linux distribution as an operating system on the server. Recently however, there has been a lively discussion in the Big Data community whether as to Amazon Linux should be recommended for users. Although Amazon Linux is specifically optimized for Amazon Web Services and it contains several useful, integral packages, including R (version 3.2.2), it is not accessible outside of Amazon cloud services. This adds considerable limitations to how and when it can actually be used. As the only way users can benefit from Amazon Linux is through AWS, this can imply additional migration costs if they want to move their operations to another cloud provider.

Secondly, and even more importantly, it is advisable that data products are developed and deployed on the same distributions to avoid bugs, time wasting, and further financial costs. It is generally recommended that, before taking your workflows or specific computations to the cloud environment, you run all the operations locally on a matching distribution. A lack of Amazon Linux release outside the AWS ecosystem therefore prevents data scientists and developers from accomplishing this goal.

In the preceding instructions, we've used Amazon Linux just for presentation purposes and its ease of deployment. However, later and in large scale development works you are more than encouraged to use other Linux distributions available through EC2 on AWS, such as CentOS or Ubuntu.

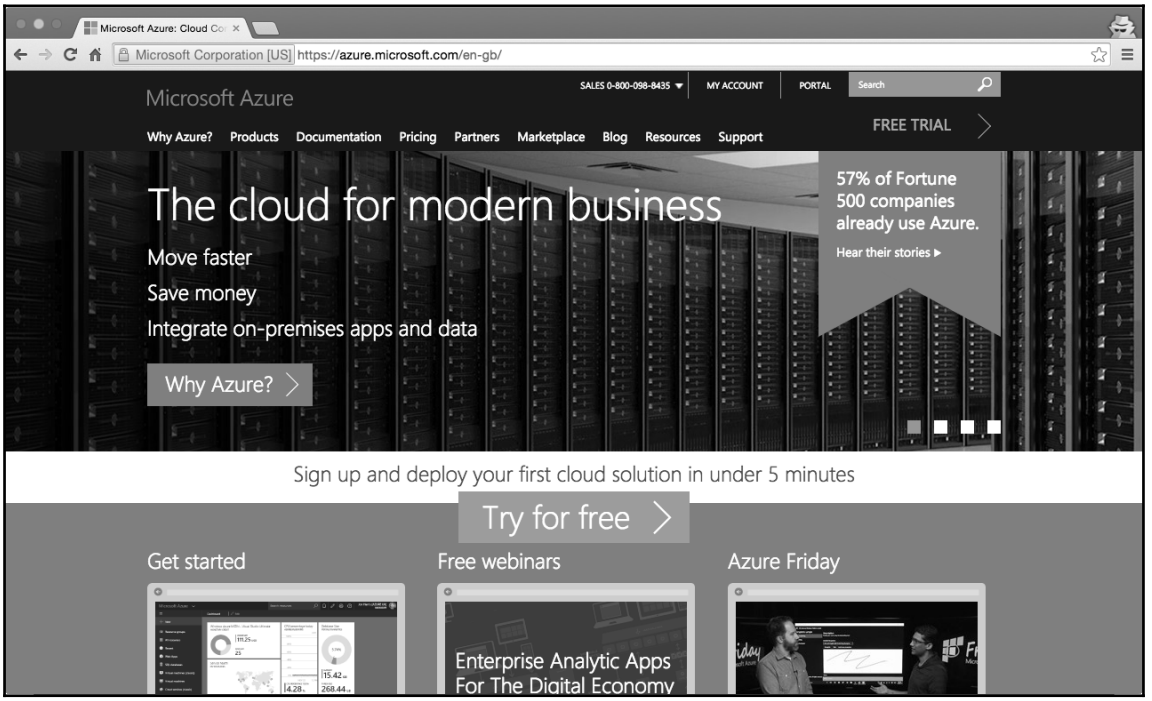
RStudio Server on Microsoft Azure virtual machines

You have already learnt how to configure and launch an Amazon EC2 instance with Amazon Linux and RStudio Server (for RedHat/CentOS) installed and ready for use for Big Data analytics and visualizations. In this section we will provide you with instructions on how to deploy an alternative virtual machine through the Microsoft Azure cloud service, but this time with the Linux Ubuntu operating system and RStudio Server (for Debian/Ubuntu) installed.

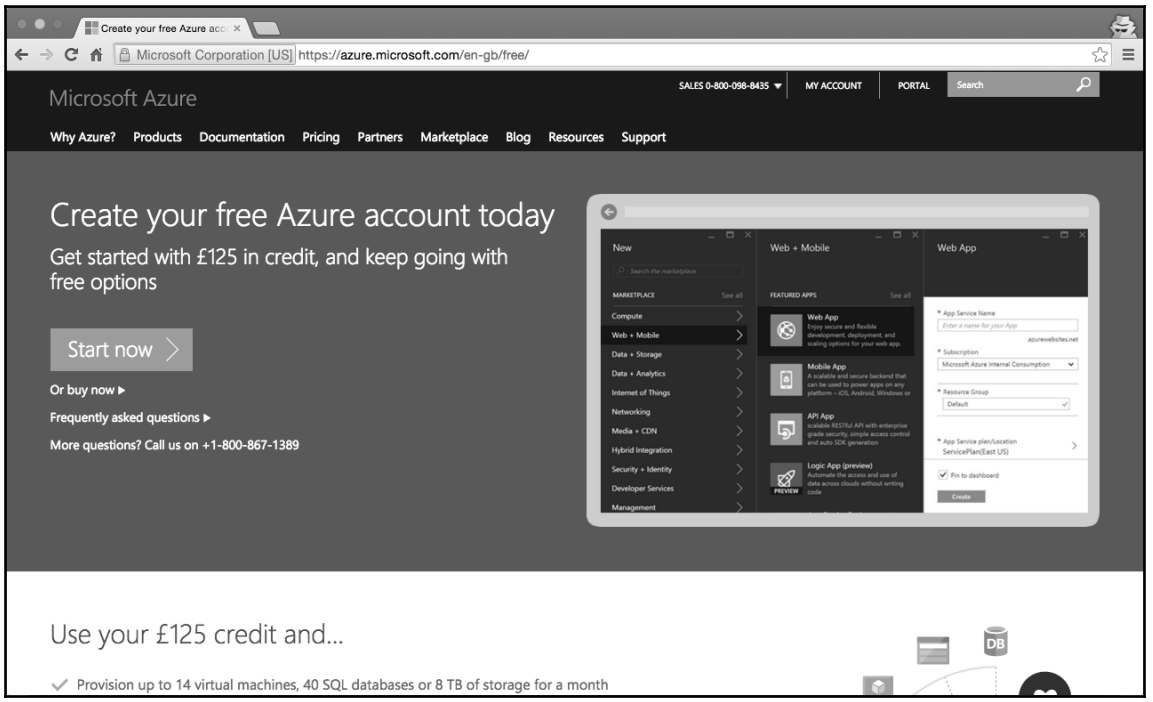
Creating a Microsoft Azure user account

In the same way that AWS offers the Free Usage Tier to new users, Microsoft Azure makes it possible for you to test its solutions with free credit to use on virtual machines, databases, and other cloud services. How much depends on your geographical location, which you need to provide when signing up to Azure. For example, as of the beginning of February 2016, users based in the United Kingdom could spend up to Â£125 of free credit on a broad range of services offered by Microsoft Azure. If you do not need to keep your services running 24/7, this allowance can actually last for quite a long time depending on which PaaS or IaaS you use. The following instructions will guide you through the process of setting up a new account on Microsoft Azure:

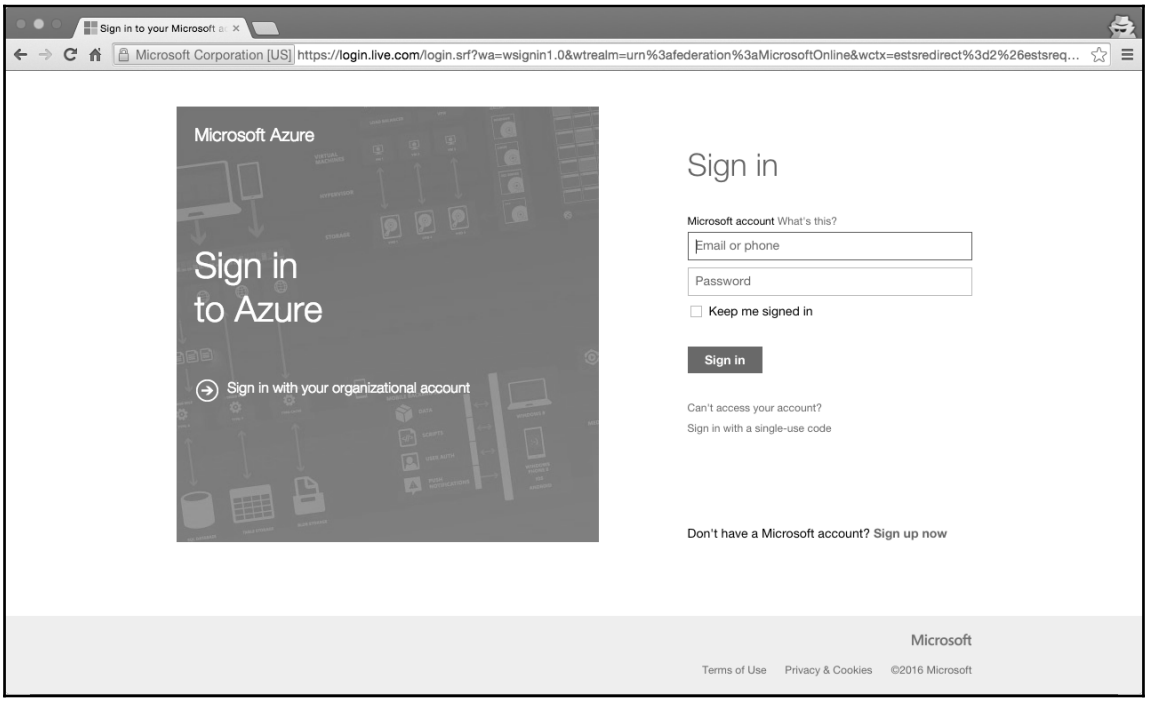
1. Go to the `https://azure.microsoft.com` website and click the **Try for free** button:



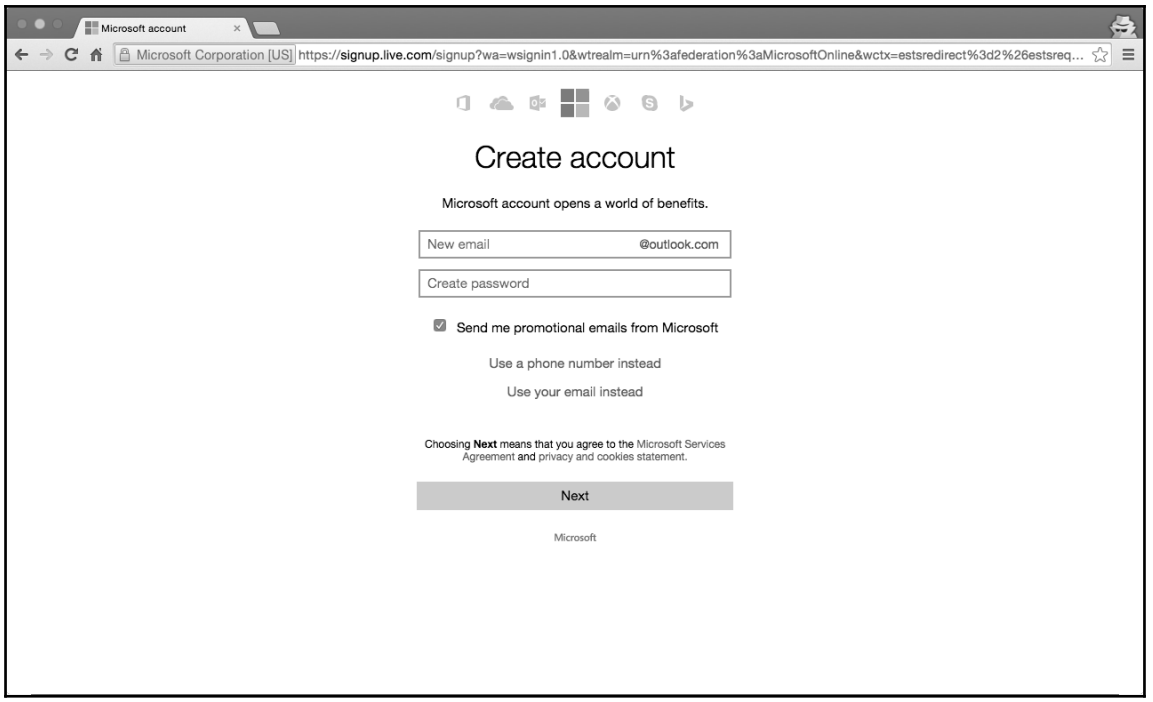
2. On the next page you will be able to read a full description of what you can get as part of your free subscription. This information may vary depending on your specific geographical location and some regions may not be eligible for any free services. Once you are happy to proceed click the **Start now** button:



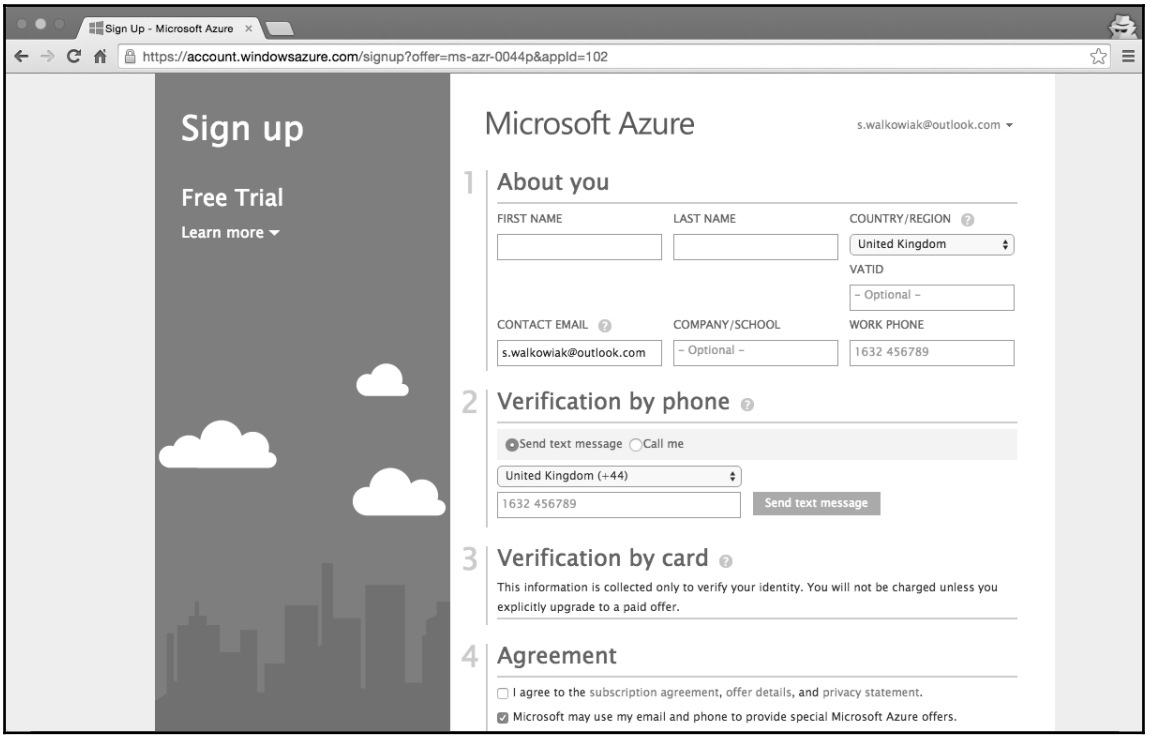
3. If you already have a Microsoft account created, you can sign-in to Azure using your existing credentials (click the **Sign in** button to proceed and skip to step 5 of this guide), otherwise click the **Sign up now** link:



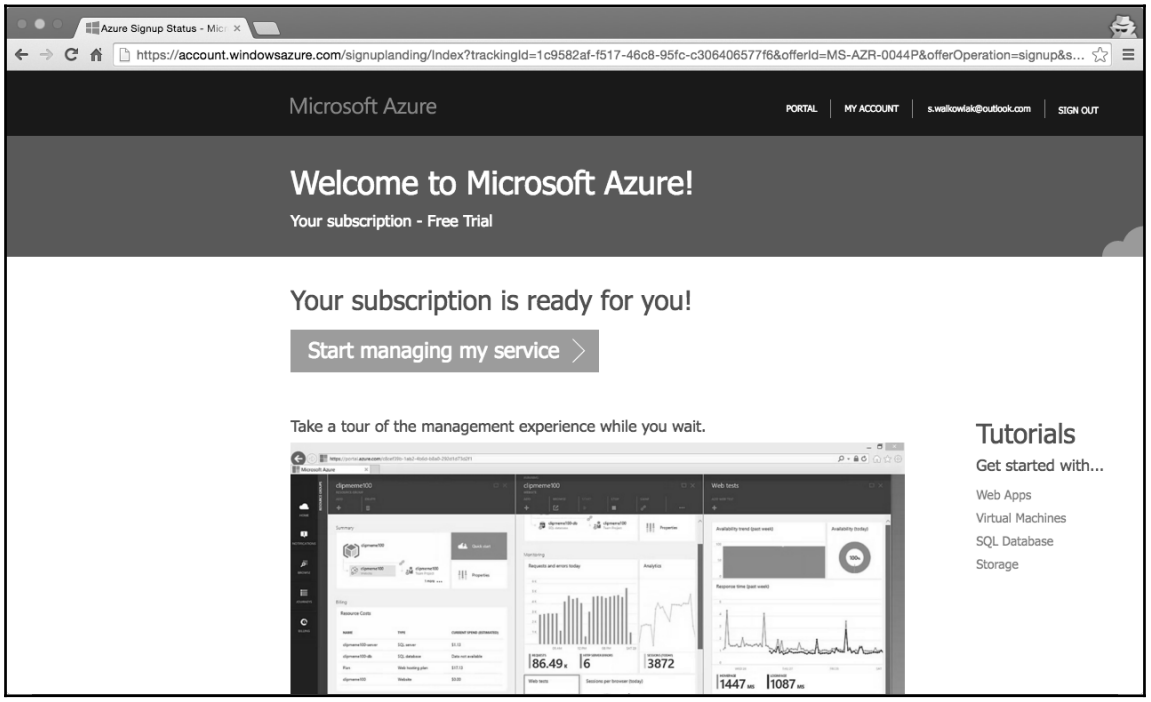
4. Create a new Microsoft account by providing a username and password. Confirm your entry by clicking the **Next** button:



5. Once you have your Microsoft account created, you will be automatically re-directed to the Azure sign up page. Complete the **About you** section and provide your mobile telephone number to verify your identity. Press **Send text message** to receive a text message with a six-digit code which you have to enter to confirm your personal telephone number. When your code is recognized you will be able to submit your credit/debit card details. Remember that your payment information will only be used to verify your identity. In the **Agreement** section tick the box to confirm that you agree to the subscription agreement, offer details, and privacy statement. Finally, you can click the **Sign up** button to proceed to Azure:

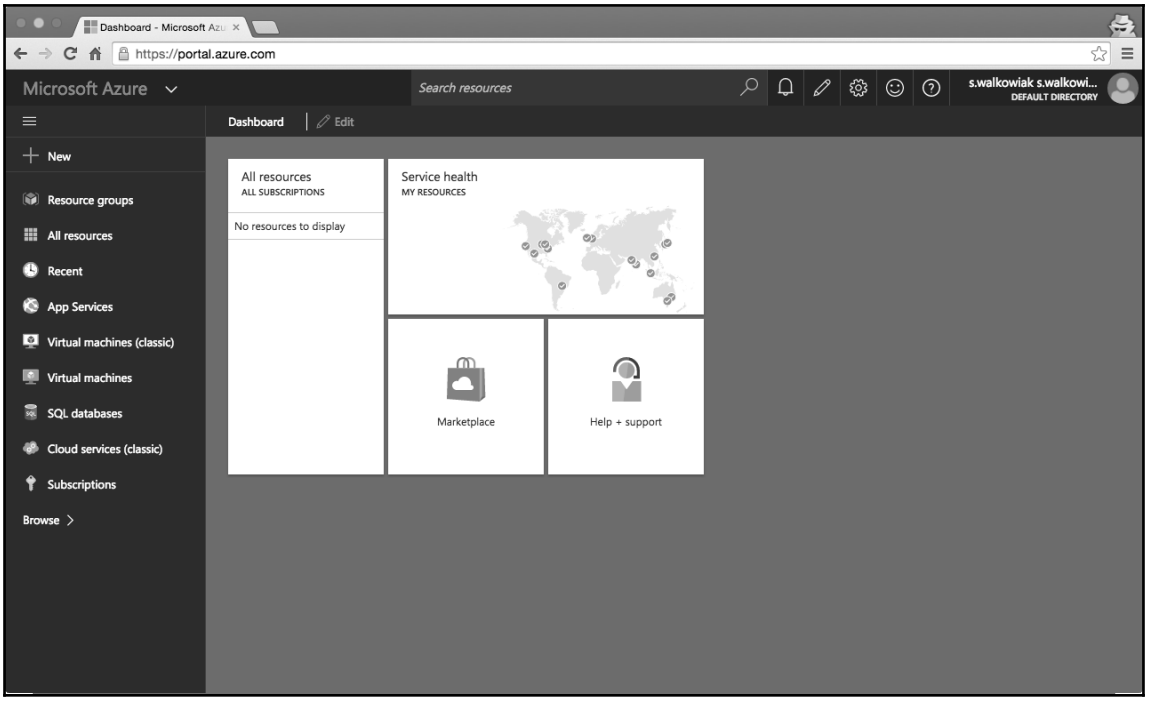


6. You will now be taken to the Microsoft Azure welcome page. It may take several minutes to configure and prepare your subscription. When it's ready you should see the following page (or similar):



Click the **Start managing my service** button to continue.

7. You have now created your Microsoft Azure account and will be re-directed to the main Azure **Dashboard** with: a top menu, which contains shortcut icons to settings, account information, notification center, dashboard customization, feedback center, and help resources; a sidebar panel with shortcuts to a variety of Azure cloud services; and the main panel which lists all available resources, displays a world map indicating current health statuses of Microsoft Azure servers, and shortcuts to the Marketplace as well as help resources.

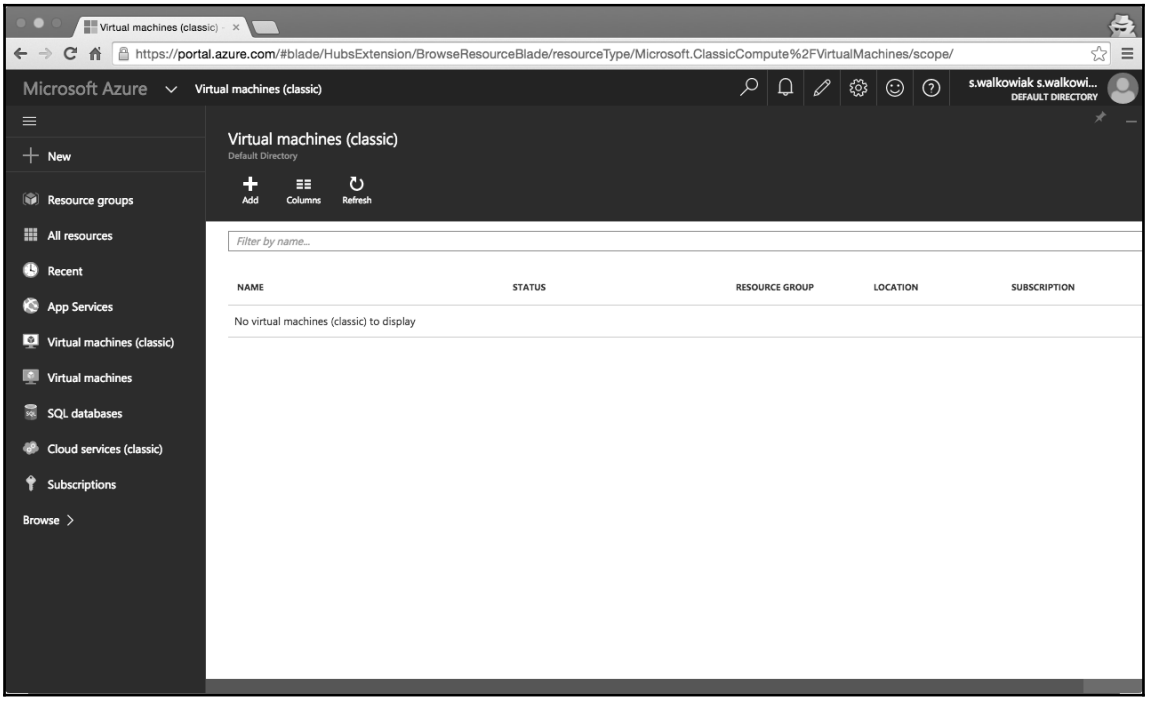


In the following section you will create your first Microsoft Azure virtual machine.

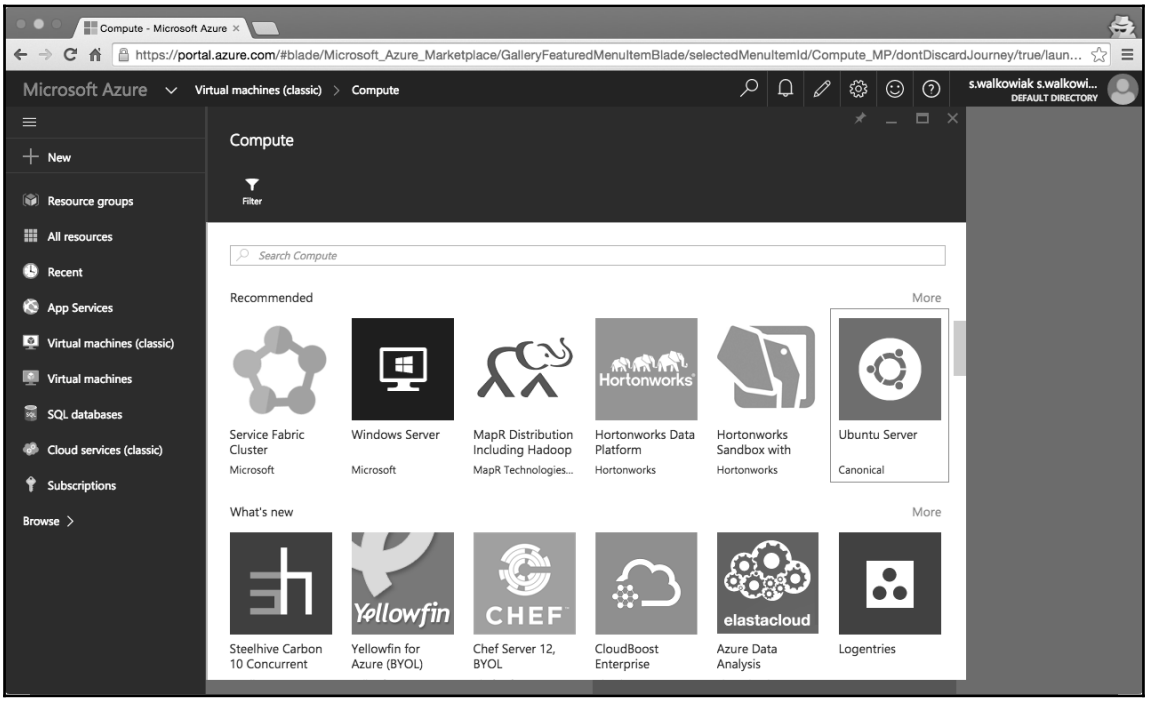
Launching your first Microsoft Azure virtual machine

To make things more interesting and your learning process more comprehensive we will set up and launch a virtual machine based on the Linux Ubuntu operating system compared to the Amazon Linux presented in the section on EC2 instances. As you are probably aware, the Amazon Linux distribution is derived from **Red Hat Enterprise Linux** (**RHEL**) and **CentOS**. Our Azure virtual machine will be based on the **Ubuntu** *flavor* of Linux which will follow a slightly different set of procedures in order to set it up:

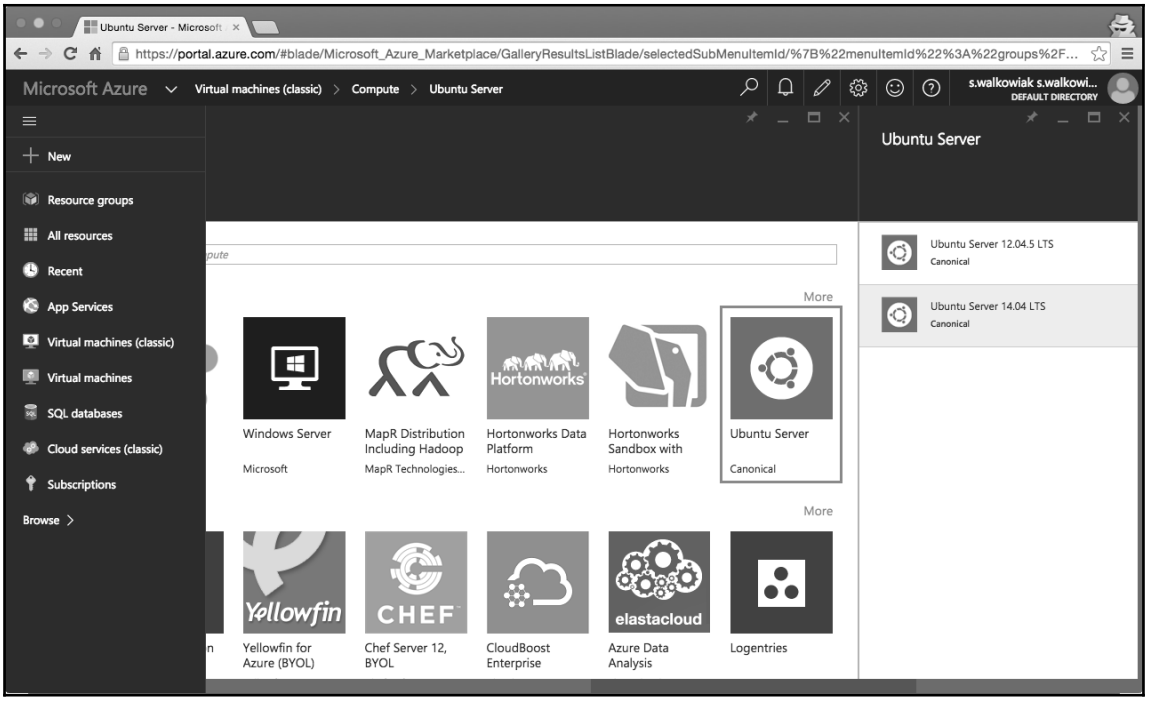
1. In the main Azure Dashboard click on the **Virtual machines (classic)** button located in the left sidebar panel. In the main panel a new page will appear. Press the **Add** button located in its top menu bar to create a new virtual machine:



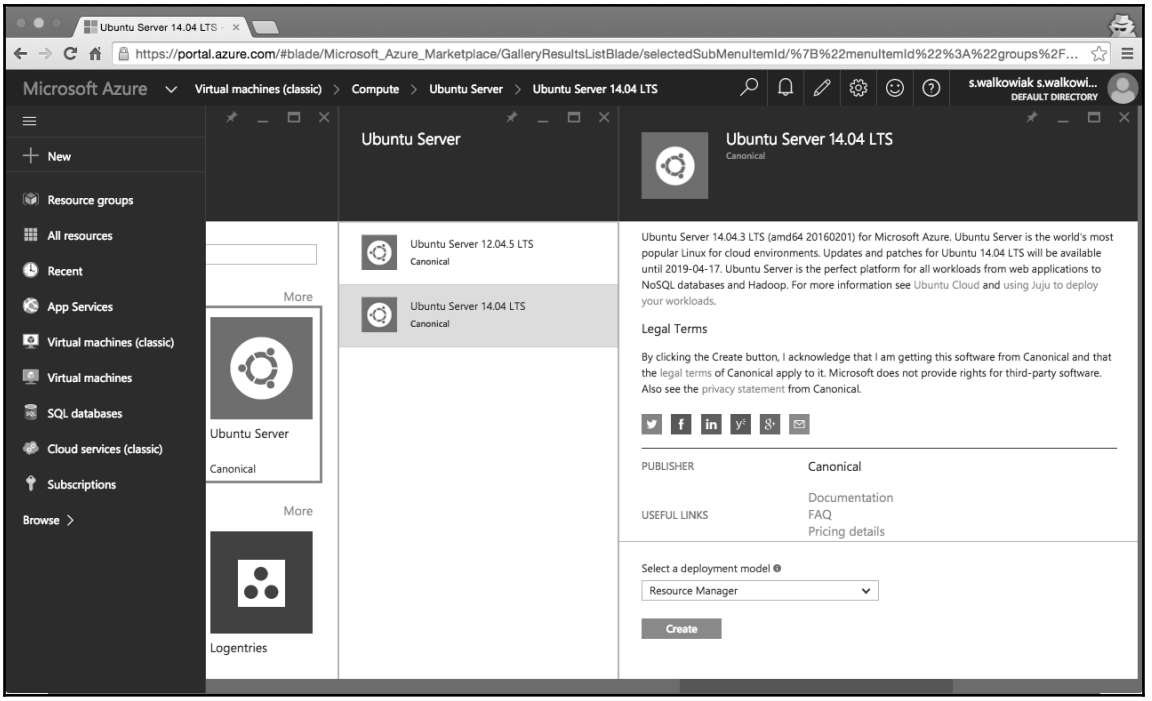
2. A number of icons representing available virtual machines will appear (if you want to inspect all of them click the **More** link). Select the Ubuntu Server as shown in the following screenshot:



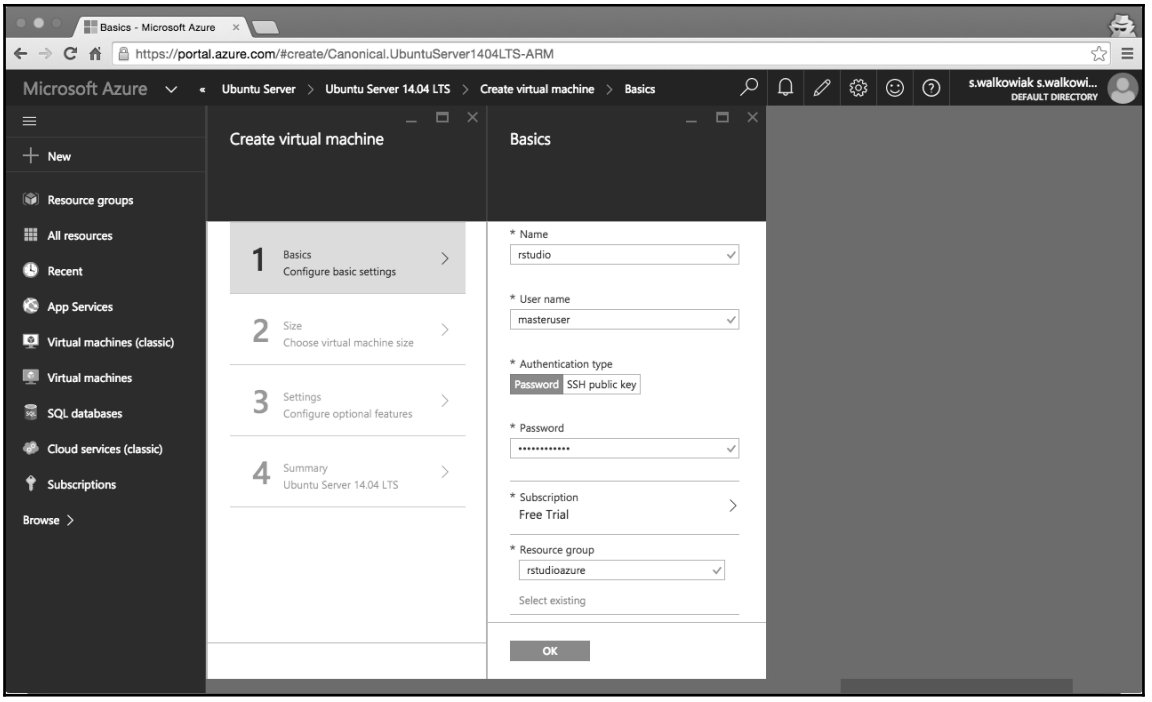
3. As of February 2016, there were two Linux Ubuntu servers available on Microsoft Azure. Choose the Ubuntu Server 14.04 LTS:



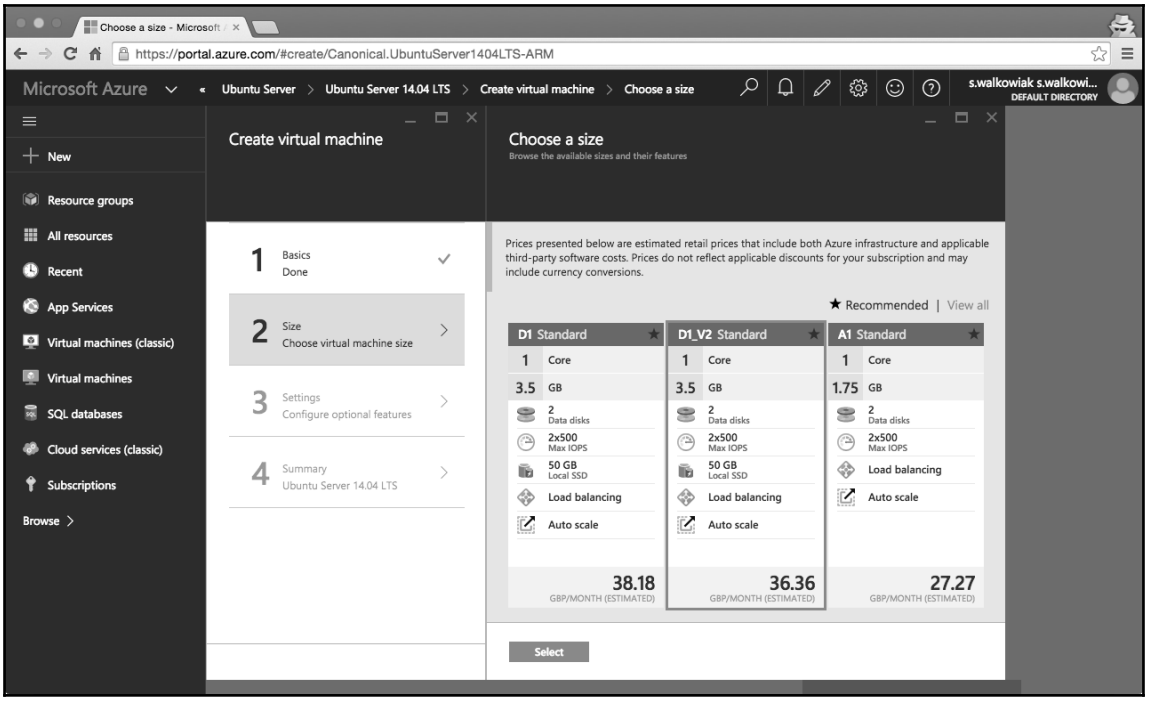
4. Read the contents of the newly created information panel and click the **Create** button to proceed:



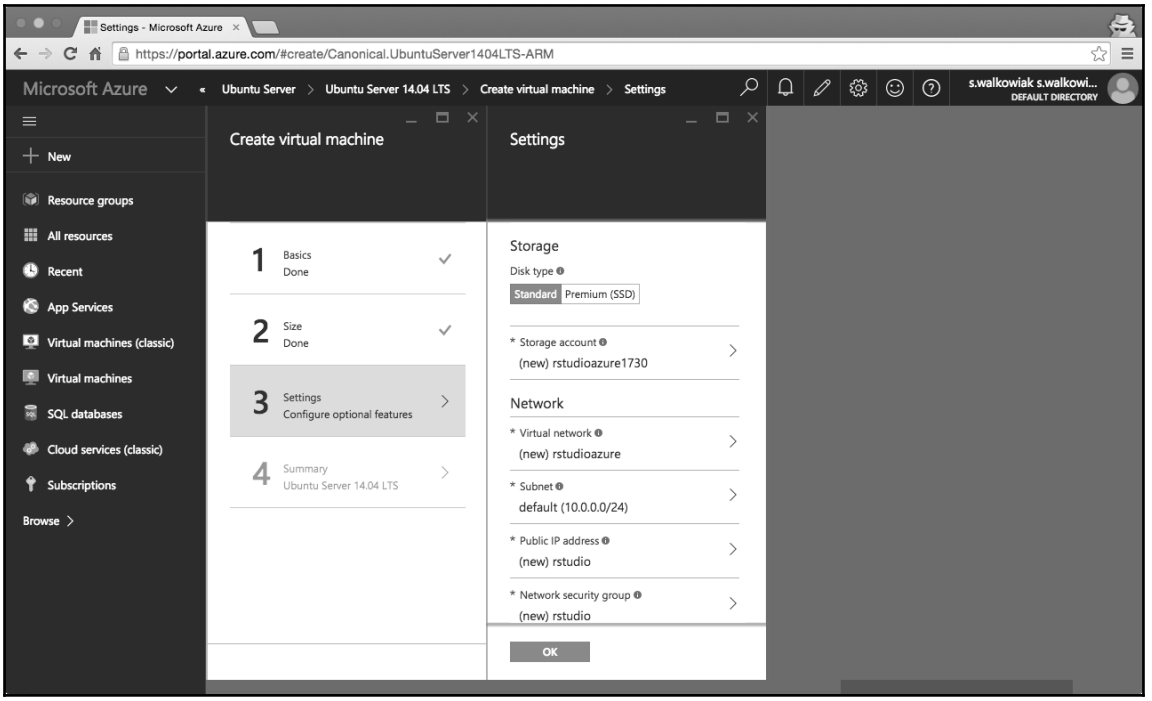
5. This will reset the contents of the main panel and now you will be able to configure the settings of the virtual machine. In the first set of options we need to define the **Name** of our machine, the **User name**, choose the **Authentication type** (we will just stick to Password for the time being), provide a valid and secure **Password**, and name of the **Resource group**. When all of this is done, click **OK**.



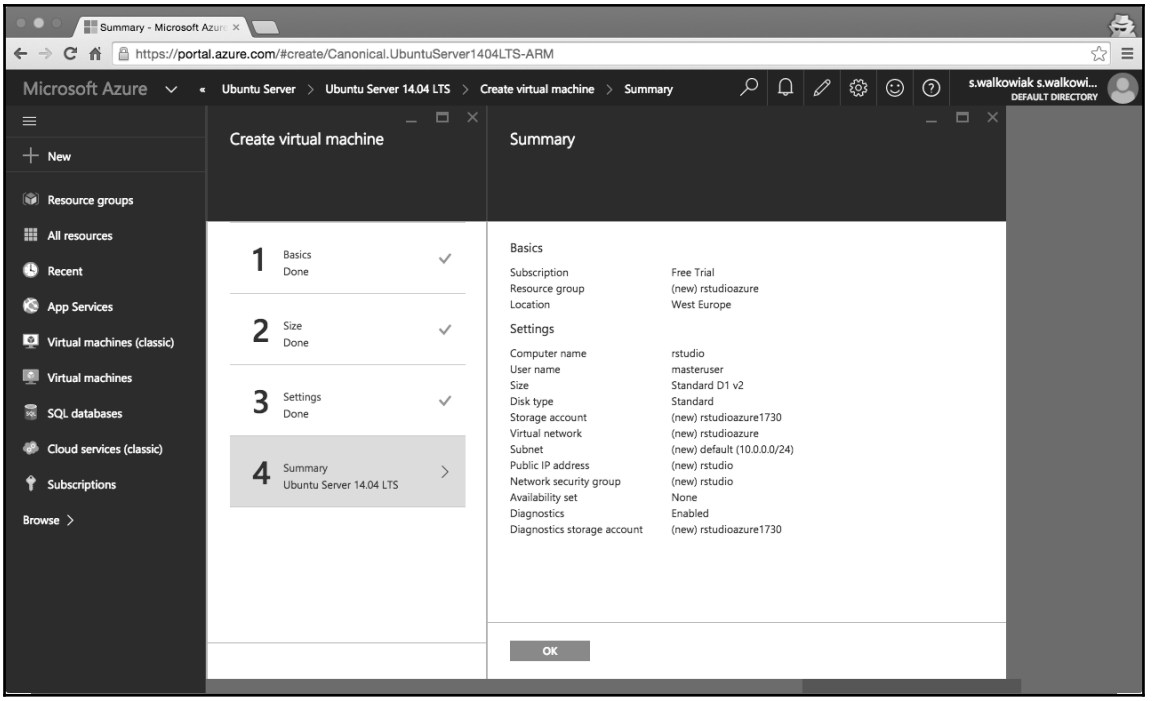
6. You will now be asked to choose the preferred size of your virtual machine. The page shows recommended choices by default, but you can see all of them by clicking the **View all** link. Below each choice, Azure calculates the estimated cost for each available machine. Remember that you won't be charged anything as long as you use it as part of your available credit. If you are not eligible for a Free Trial offer, the figures are only rough estimates of the charges and may increase or decrease depending on the specific configuration of your virtual machine. We will choose the **D1_V2 Standard** type of machine, which includes 1 virtual CPU, 3.5 GB of RAM, two data disks and an additional storage of 50 GB on a solid state disk. Confirm your selection by clicking the **Select** button:



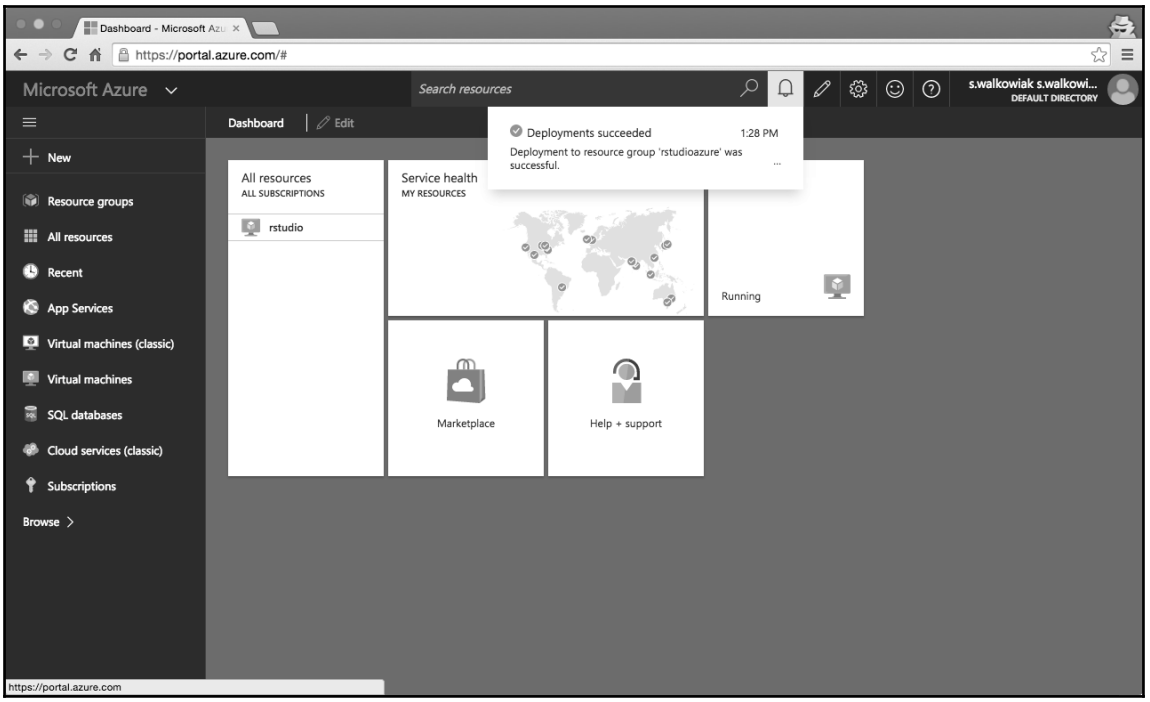
7. The **Settings** are created automatically and, as long as you don't change anything you may simply continue to the final summary view by clicking the **OK** button.



8. The summary section confirms the configuration of your virtual machine. Press **OK** to continue.



9. Azure will now initialize deployment of your virtual machine. It may take several minutes. You will receive a notification message when it becomes ready and your virtual machine will be visible in the available resources on your main Dashboard screen:



This completes the creation and deployment of a Microsoft Azure virtual machine. In the next section we will install an appropriate distribution of RStudio Server.

Installing RStudio Server on a Microsoft Azure virtual machine

The procedures involved in the installation of RStudio Server on a Microsoft Azure Linux Ubuntu machine are just a little bit trickier than on an Amazon Linux EC2 instance, but users will benefit from much more flexibility and control over their machine:

1. Using the SSH client and credentials used to launch the Azure instance, connect to your virtual machine:

```
ssh masteruser@13.81.8.124
```

   You will be asked for your password to verify administrator rights. If correctly verified, you will be allowed to work on the machine and will see a welcome output.

2. In order to install RStudio Server, the machine needs to contain the most recent version of core R – just like in the Amazon EC2 example earlier in this chapter. To obtain the latest R packages for Ubuntu Trusty 14.04 LTS we need to set a CRAN mirror as follows:

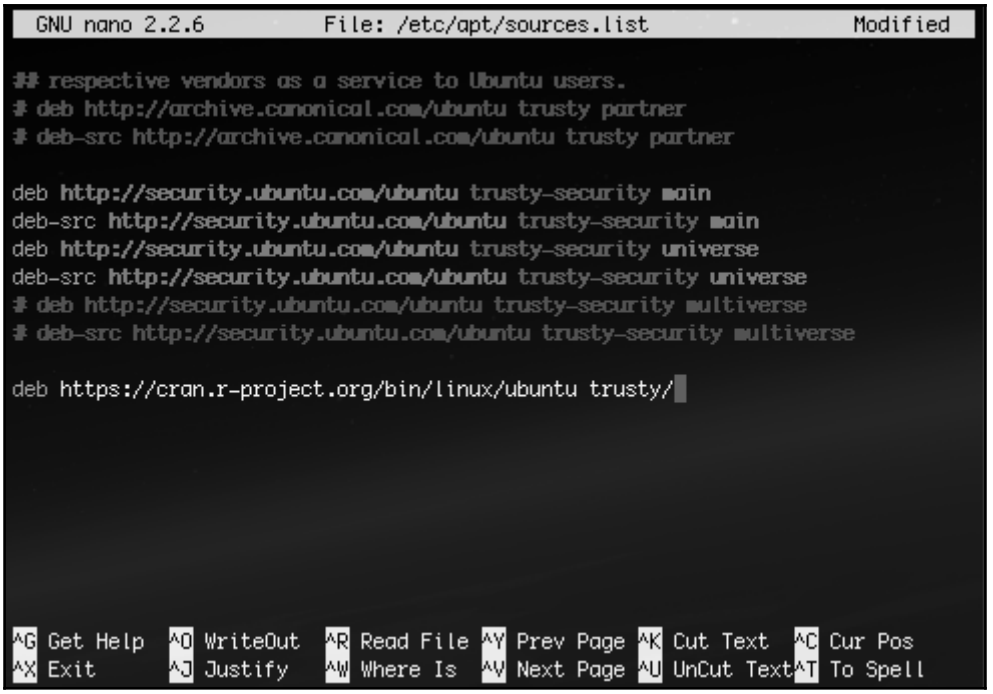
```
deb https://cran.r-project.org/bin/linux/ubuntu trusty/
```

   If the `deb` command is not found, open the GNU Nano editor for Ubuntu with the `sources.list` file:

```
sudo nano /etc/apt/sources.list
```

Scroll down and add the following line to the list of accepted sources:

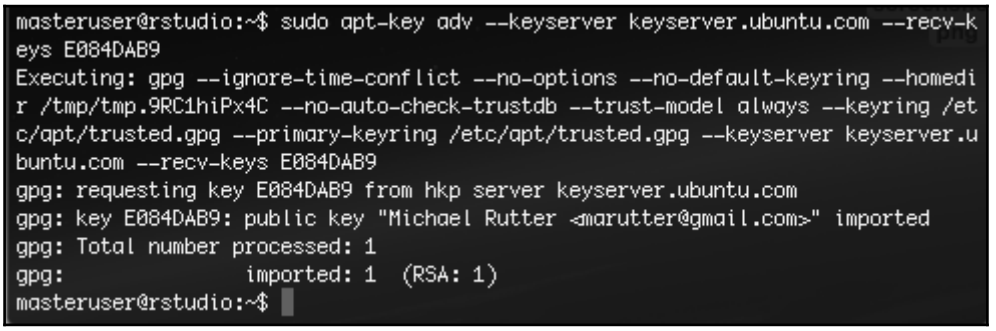
```
deb https://cran.r-project.org/bin/linux/ubuntu trusty/
```



Save the changes by pressing *Ctrl* + *O* and exit the Nano editor by pressing *Ctrl* + *X*.

3. The Ubuntu archives on CRAN are signed with the key ID E084DAB9. To add the key to your system, use the following command:

```
sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys E084DAB9
```



Then feed it to `apt-key` with:

```
gpg -a --export E084DAB9 | sudo apt-key add -
```

4. The Ubuntu r-base and recommended R packages are installed into the directory: `/usr/lib/R/library`. They can be updated and upgraded using `apt-get` with:

```
sudo apt-get update
sudo apt-get upgrade
```

5. We can now install core R as follows:

```
sudo apt-get install r-base
```

This will install the most recent release of the base R.

6. We may now download and install RStudio Server for Linux Debian/Ubuntu according to the instructions provided at: `https://www.rstudio.com/products/rstudio/download-server/`:

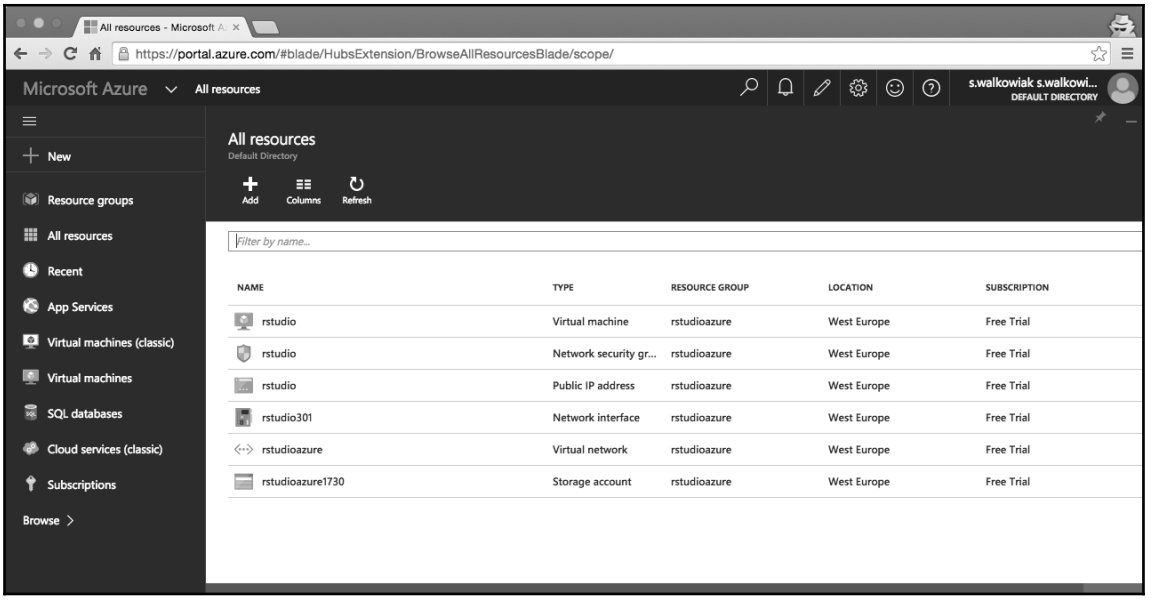
```
sudo apt-get install gdebi-core
...
wget https://download2.rstudio.org/rstudio-server-0.99.879-amd64.deb
...
sudo gdebi rstudio-server-0.99.879-amd64.deb
```

7. Finally, we can add users and specify their UNIX passwords using the command below:

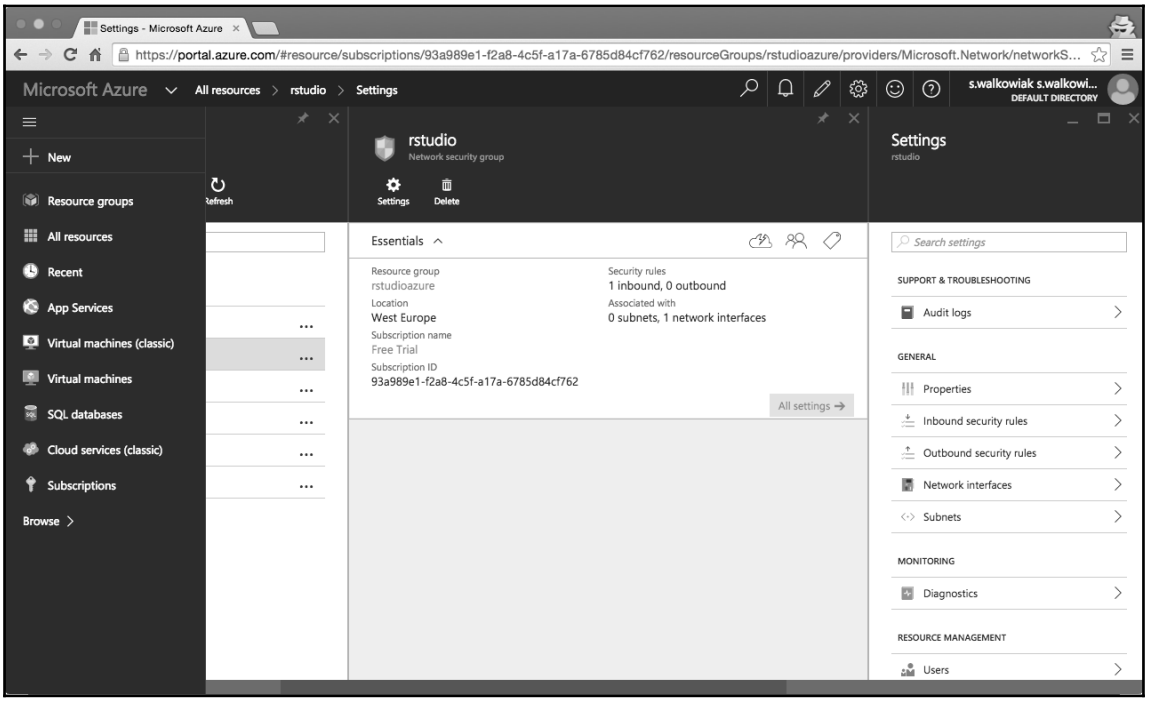
```
sudo adduser user01
```

   Here you will be asked to enter a new UNIX password for `user01`-you can type whatever you like. You can also add other information about this and other users.

8. The R and RStudio Server installations are now complete. At this stage we just need to add port `8787` to our security group's rules on the created virtual machine. Once logged-in to the Microsoft Azure platform, click on the **Resource groups** in the left sidebar panel. This will invoke a new page in the main panel with all of the services we have enabled for our machine:

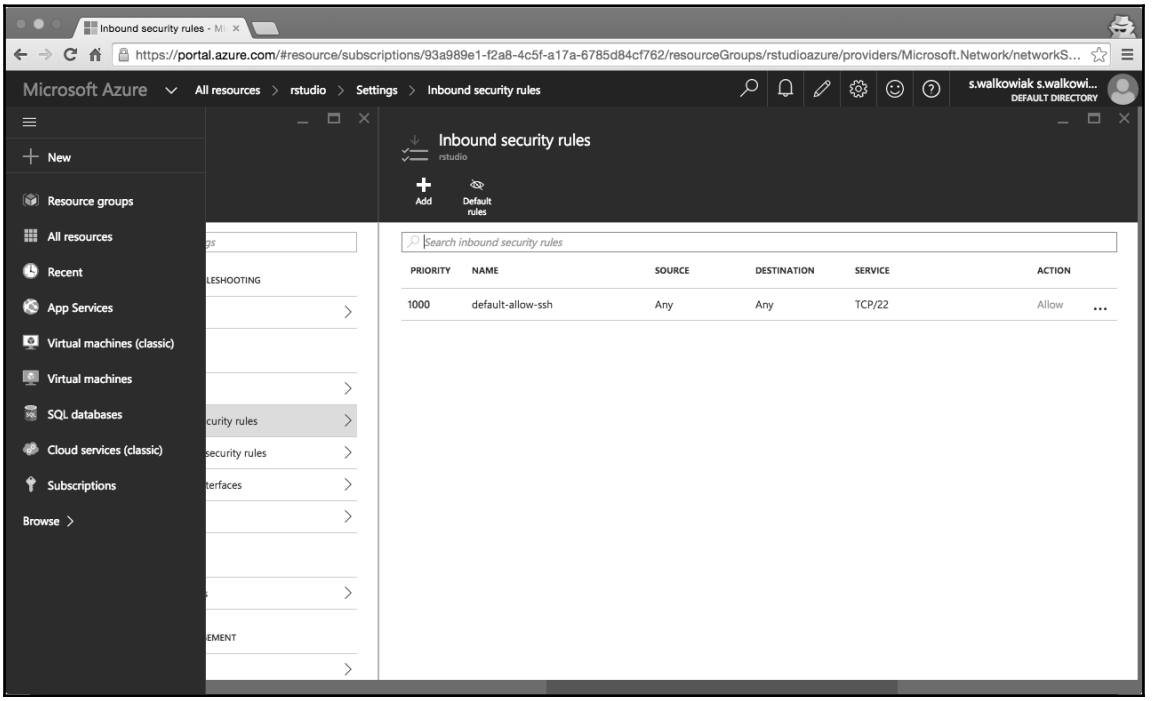


9. Click on the **rstudio** (**NAME** column) next to **Network security group** (**TYPE** column). This will open a new sub-page with essential network security group details as shown in the following screenshot:

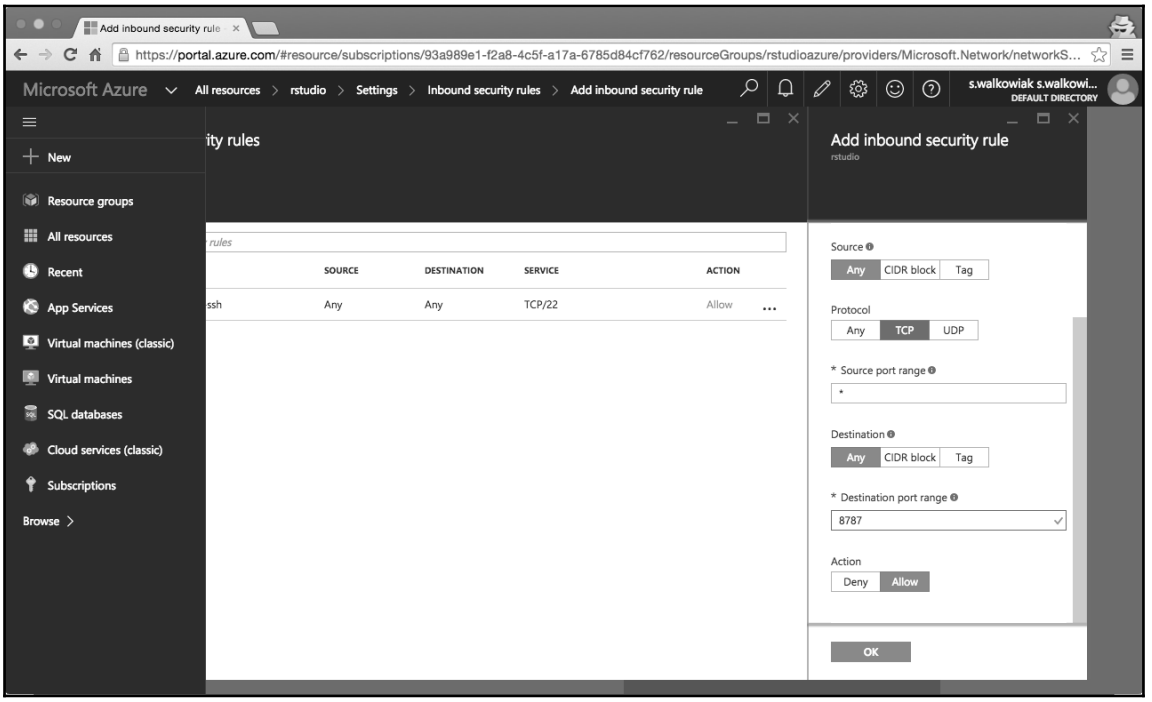


Note that, currently, we have only one inbound security rule set up. We need to create another one by adding port `8787`. Click the **All settings** link to proceed.

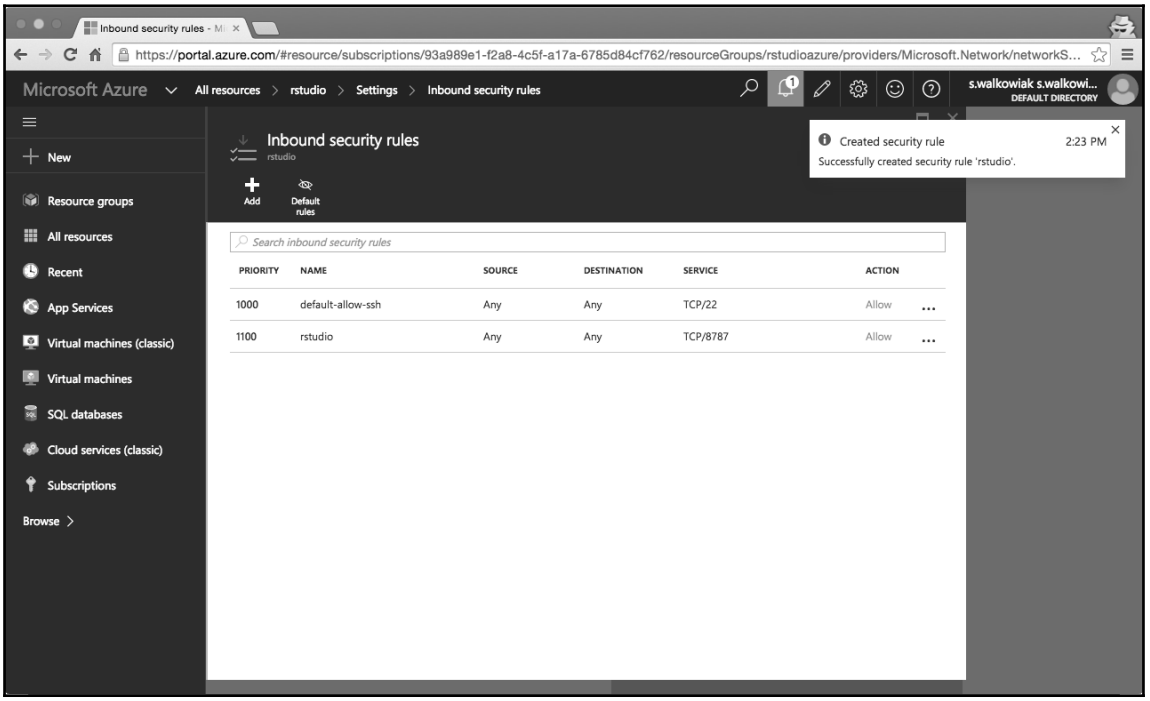
10. From the**Settings,** select**Inbound security rules** and click the **Add** icon above:



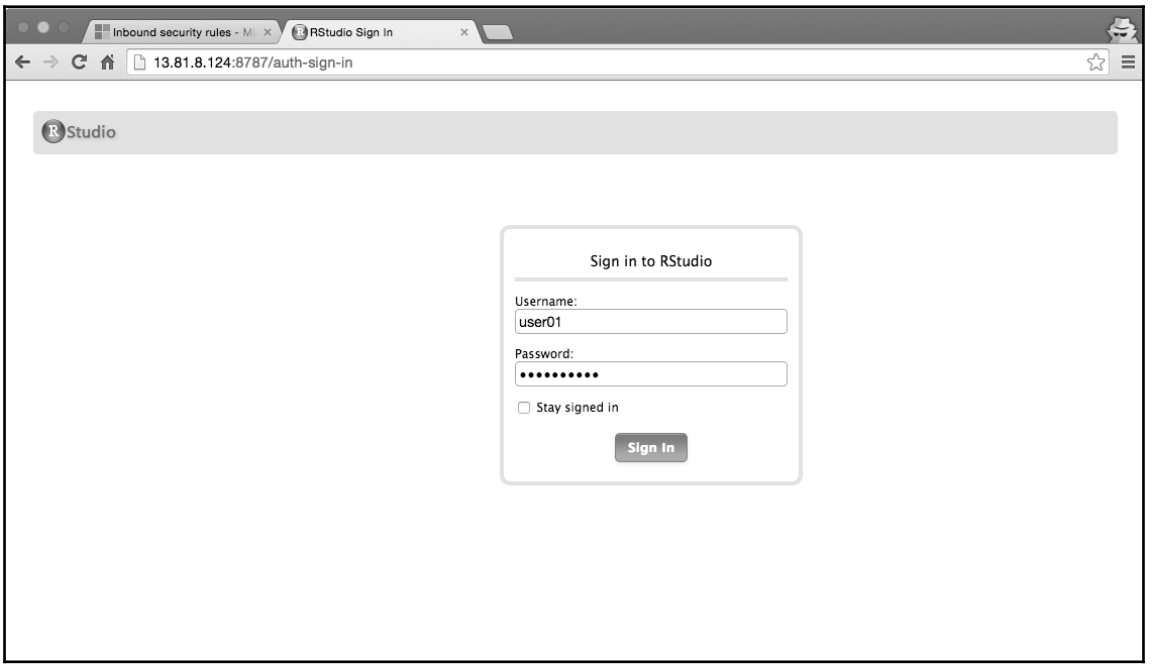
11. Now we can add our additional inbound security rule. Let's call it `rstudio`, we will keep its default priority set to `1100`, click on **TCP** to choose it as our **Protocol** and set the **Destination port range** to `8787`. Make sure that the highlighted **Action** is **Allow**. Accept all changes by clicking **OK**:



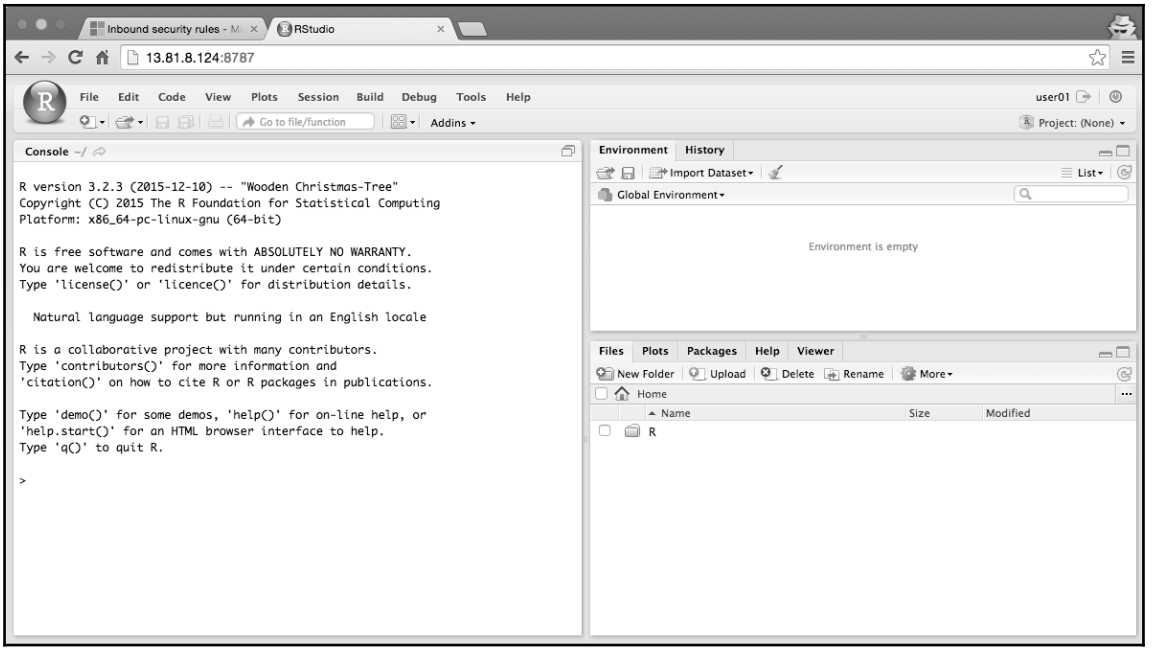
After a few seconds you will get a confirmation notification and the new rule will have been added to the existing list of inbound security rules as shown in the following screenshot:



12. Finally, navigate your web browser to `http://IP:8787` (where `IP` is the IP address of your virtual machine) and input your credentials for a specific user to login to the RStudio Server session:



13. A few seconds later you will be able to start using RStudio Server in the normal way:



This completes the installation of R and RStudio Server on a Microsoft Azure virtual machine with Linux Ubuntu 14.04 LTS version. When all your processing work is done make sure to STOP the virtual machine from the Azure Dashboard in order to de-allocate it and avoid any future charges. Note that all your files stored on the de-allocated storage disk attached to the virtual machine will be deleted (but not the installed applications). The de-allocated virtual machine can be resumed in the future, but it will operate with a different hostname/IP address.

Summary

In this chapter, we have introduced you to the concept of cloud computing and its advantages for *Big*, *Fast*, and *Smart* data analytics.

The initial section has provided you with a broad outline of cloud services offered by leading cloud computing vendors: Amazon Web Services, Microsoft Azure, and Google Chrome Platform.

The second part of the chapter focused on practical skills which allow you to create free and trial user accounts on Amazon Web Services and Microsoft Azure and to launch fully-operational virtual machines built on Amazon Linux and Linux Ubuntu 14.04 LTS distributions with base R and the most recent RStudio Server statistical packages.

The skills learned in this chapter will be used by you throughout the remainder of the book. In *Chapter 5*, *Hadoop and MapReduce Framework for R*, you will learn how to set up a **Hadoop cluster** on your virtual machine and how to use **Hadoop Distributed File System** and **MapReduce** frameworks directly from RStudio Server for Big Data processing and analysis.