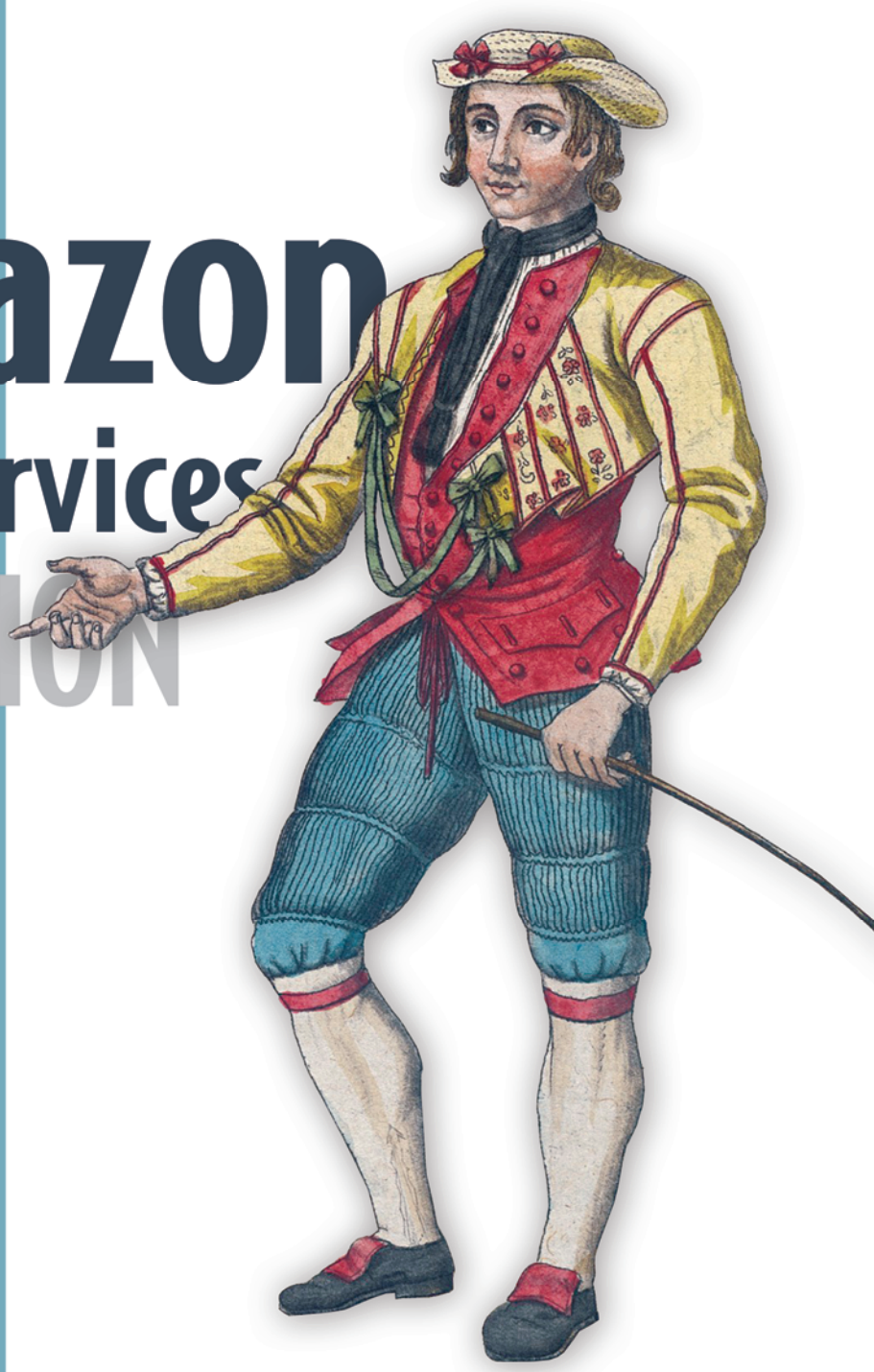
Amazon Web Services IN ACTION

SECOND EDITION

Michael Wittig
Andreas Wittig

Foreword by Ben Whaley

SAMPLE CHAPTER



MANNING

Amazon Web Services in Action, Second Edition

by Michael Wittig
and Andreas Wittig

Chapter 1

brief contents

Part 1

Getting started

Have you watched a blockbuster on Netflix, bought a gadget on Amazon.com, or booked a room on Airbnb today? If so, you have used Amazon Web Services (AWS) in the background. Because Netflix, Amazon.com, and Airbnb all use Amazon Web Services for their business.

Amazon Web Services is the biggest player in the cloud computing markets. According to analysts, AWS maintains a market share of more than 30%.[1] Another impressive number: AWS reported net sales of $4.1 billion USD for the quarter ending in June 2017.[2] AWS data centers are distributed worldwide in North America, South America, Europe, Asia, and Australia. But the cloud does not consist of hardware and computing power alone. Software is part of every cloud platform and makes the difference for you, as a customer who aims to provide a valuable experience to your services's users. The research firm Gartner has yet again classified AWS as a leader in their Magic Quadrant for Cloud Infrastructure as a Service in 2017. Gartner's Magic Quadrant groups vendors into four quadrants: niche players, challengers, visionaries, and leaders, and provides a quick overview of the cloud computing market.[3] Being recognized as a leader attests AWS's high speed and high quality of innovation.

---

[1] Synergy Research Group, "The Leading Cloud Providers Continue to Run Away with the Market," http://mng.bz/qDYo.

[2] Amazon, 10-Q for Quarter Ended June 30 (2017), http://mng.bz/1LAX.

[3] AWS Blog, "AWS Named as a Leader in Gartner's Infrastructure as a Service (IaaS) Magic Quadrant for 7th Consecutive Year," http://mng.bz/0W1W.

The first part of this book will guide you through your initial steps with AWS. You will get an impression of how you can use AWS to improve your IT infrastructure.

Chapter 1 introduces cloud computing and AWS. This will get you familiar with the big-picture basics of how AWS is structured.

Chapter 2 brings Amazon Web Service into action. Here, you will spin up and dive into a complex cloud infrastructure with ease.

1 What is Amazon Web Services?

This chapter covers

- Overview of Amazon Web Services
- The benefits of using Amazon Web Services
- What you can do with Amazon Web Services
- Creating and setting up an AWS account

Amazon Web Services (AWS) is a platform of web services that offers solutions for computing, storing, and networking, at different layers of abstraction. For example, you can use block-level storage (a low level of abstraction) or a highly distributed object storage (a high level of abstraction) to store your data. You can use these services to host websites, run enterprise applications, and mine tremendous amounts of data. *Web services* are accessible via the internet by using typical web protocols (such as HTTP) and used by machines or by humans through a UI. The most prominent services provided by AWS are EC2, which offers virtual machines, and S3, which offers storage capacity. Services on AWS work well together: you can use them to replicate your existing local network setup, or you can design a new setup from scratch. The pricing model for services is pay-per-use.

As an AWS customer, you can choose among different *data centers*. AWS data centers are distributed worldwide. For example, you can start a virtual machine in Japan in exactly the same way as you would start one in Ireland. This enables you to serve customers worldwide with a global infrastructure.

The map in figure 1.1 shows AWS's data centers. Access is limited to some of them: some data centers are accessible for U.S. government organizations only, and special conditions apply for the data centers in China. Additional data centers have been announced for Bahrain, Hong Kong, Sweden, and the U.S..

* Limited access

Figure 1.1 AWS data center locations

In more general terms, AWS is known as a *cloud computing platform*.

1.1 What is cloud computing?

Almost every IT solution is labeled with the term *cloud computing* or just *cloud* nowadays. Buzzwords like this may help sales, but they're hard to work with in a book. So for the sake of clarity, let's define some terms.

Cloud computing, or the cloud, is a metaphor for supply and consumption of IT resources. The IT resources in the cloud aren't directly visible to the user; there are layers of abstraction in between. The level of abstraction offered by the cloud varies, from offering virtual machines (VMs) to providing software as a service (SaaS) based on complex distributed systems. Resources are available on demand in enormous quantities, and you pay for what you use.

The official definition from the National Institute of Standards and Technology:

> *Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (networks, virtual machines, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.*
>
> —National Institute of Standards and Technology, *The NIST Definition of Cloud Computing*

Clouds are often divided into three types:

- *Public*—A cloud managed by an organization and open to use by the general public.
- *Private*—A cloud that virtualizes and distributes the IT infrastructure for a single organization.
- *Hybrid*—A mixture of a public and a private cloud.

AWS is a public cloud. Cloud computing services also have several classifications:

- *Infrastructure as a service (IaaS)*—Offers fundamental resources like computing, storage, and networking capabilities, using virtual machines such as Amazon EC2, Google Compute Engine, and Microsoft Azure.
- *Platform as a service (PaaS)*—Provides platforms to deploy custom applications to the cloud, such as AWS Elastic Beanstalk, Google App Engine, and Heroku.
- *Software as a service (SaaS)*—Combines infrastructure and software running in the cloud, including office applications like Amazon WorkSpaces, Google Apps for Work, and Microsoft Office 365.

The AWS product portfolio contains IaaS, PaaS, and SaaS. Let's take a more concrete look at what you can do with AWS.

1.2 What can you do with AWS?

You can run all sorts of application on AWS by using one or a combination of services. The examples in this section will give you an idea of what you can do.

1.2.1 Hosting a web shop

John is CIO of a medium-sized e-commerce business. He wants to develop a fast and reliable web shop. He initially decided to host the web shop on-premises, and three years ago he rented machines in a data center. A web server handles requests from customers, and a database stores product information and orders. John is evaluating how his company can take advantage of AWS by running the same setup on AWS, as shown in figure 1.2.

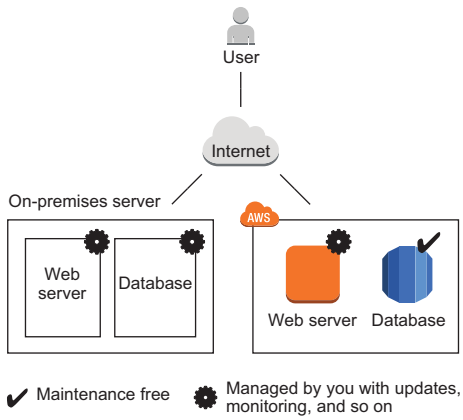


Figure 1.2 Running a web shop on-premises vs. on AWS

John not only wants to lift-and-shift his current on-premises infrastructure to AWS; he wants to get the most out of the advantages the cloud is offering. Additional AWS services allow John to improve his setup.

- The web shop consists of dynamic content (such as products and their prices) and static content (such as the company logo). Splitting these up would reduce the load on the web servers and improve performance by delivering the static content over a content delivery network (CDN).
- Switching to maintenance-free services including a database, an object store, and a DNS system would free John from having to manage these parts of the system, decreasing operational costs and improving quality.
- The application running the web shop can be installed on virtual machines. Using AWS, John can run the same amount of resources he was using on his on-premises machine, but split into multiple smaller virtual machines at no extra cost. If one of these virtual machines fails, the load balancer will send customer requests to the other virtual machines. This setup improves the web shop's reliability.

Figure 1.3 shows how John enhanced the web shop setup with AWS.

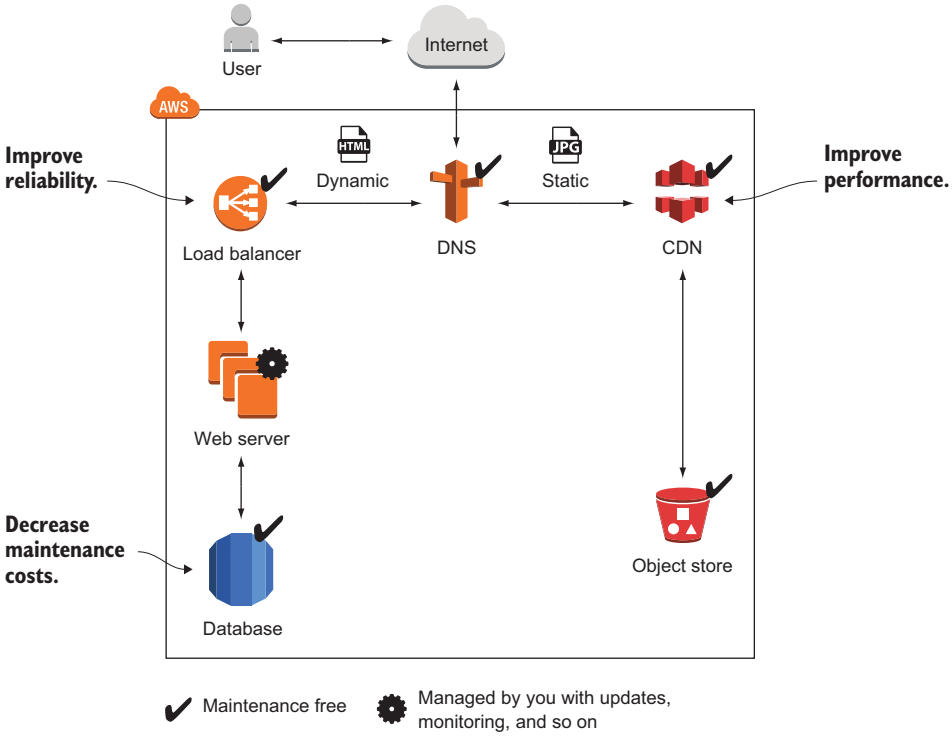


Figure 1.3 Running a web shop on AWS with CDN for better performance, a load balancer for high availability, and a managed database to decrease maintenance costs

John is happy with running his web shop on AWS. By migrating his company's infrastructure to the cloud, he was able to increase the reliability and performance of the web shop.

1.2.2 Running a Java EE application in your private network

Maureen is a senior system architect in a global corporation. She wants to move parts of her company's business applications to AWS when the data-center contract expires in a few months, to reduce costs and gain flexibility. She wants to run enterprise applications (such as Java EE applications) consisting of an application server and an SQL database on AWS. To do so, she defines a virtual network in the cloud and connects it to the corporate network through a Virtual Private Network (VPN) connection. She installs application servers on virtual machines to run the Java EE application. Maureen also wants to store data in an SQL database service (such as Oracle Database Enterprise Edition or Microsoft SQL Server EE).

For security, Maureen uses subnets to separate systems with different security levels from each other. By using access-control lists, she can control ingoing and outgoing traffic for each subnet. For example, the database is only accessible from the JEE server's subnet which helps to protect mission-critical data. Maureen controls traffic to the internet by using Network Address Translation (NAT) and firewall rules as well. Figure 1.4 illustrates Maureen's architecture.

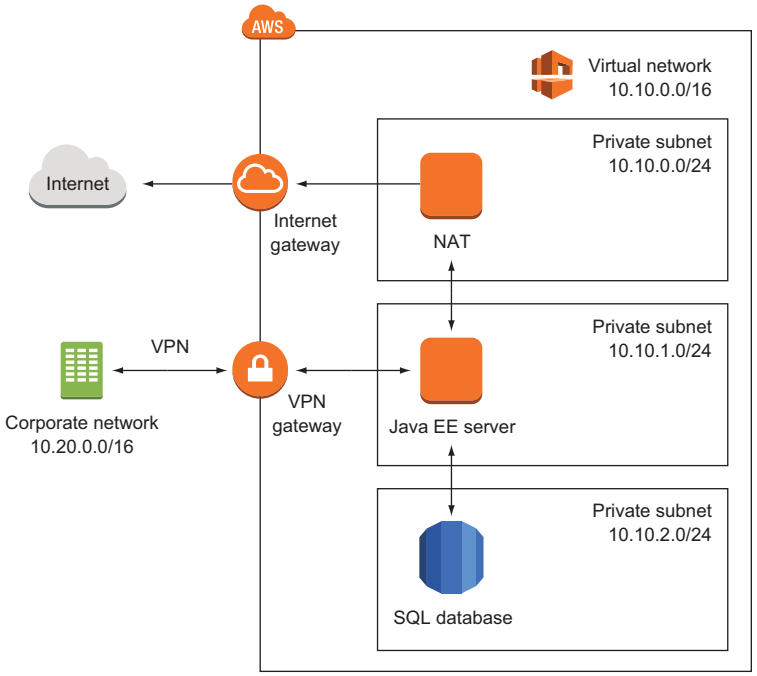


Figure 1.4 Running a Java EE application with enterprise networking on AWS improves flexibility and lowers costs.

Maureen has managed to connect the local data center with a private network running remotely on AWS to enable clients to access the JEE server. To get started, Maureen uses a VPN connection between the local data center and AWS, but she is already thinking about setting up a dedicated network connection to reduce network costs and increase network throughput in the future.

The project was a great success for Maureen. She was able to reduce the time needed to set up an enterprise application from months to hours, as AWS can take care of the virtual machines, databases, and even the networking infrastructure on demand within a few minutes. Maureen's project also benefits from lower infrastructure costs on AWS, compared to using their own infrastructure on-premises.

1.2.3 Implementing a highly available system

Alexa is a software engineer working for a fast-growing startup. She knows that Murphy's Law applies to IT infrastructure: anything that can go wrong *will* go wrong. Alexa is working hard to build a highly available system to prevent outages from ruining the business. All services on AWS are either highly available or can be used in a highly available way. So, Alexa builds a system like the one shown in figure 1.5 with a high availability architecture. The database service is offered with replication and fail-over handling. In case the master database instance fails, the standby database is promoted as the new master database automatically. Alexa uses virtual machines acting as web servers. These virtual machines aren't highly available by default, but Alexa launches multiple virtual machines in different data centers to achieve high availability. A load balancer checks the health of the web servers and forwards requests to healthy machines.

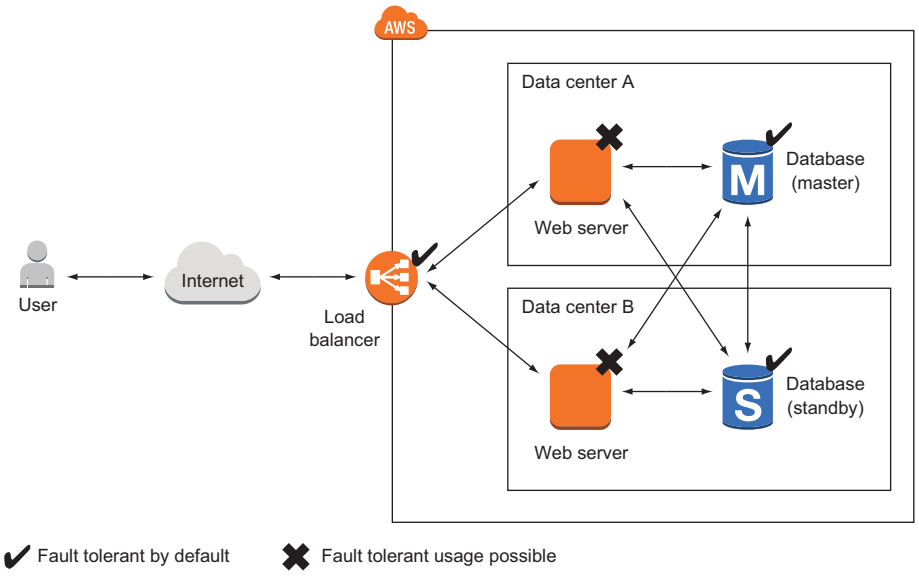


Figure 1.5 Building a highly available system on AWS by using a load balancer, multiple virtual machines, and a database with master-standby replication

So far, Alexa has protected the startup from major outages. Nevertheless, she and her team are always planning for failure and are constantly improving the resilience of their systems.

1.2.4 Profiting from low costs for batch processing infrastructure

Nick is a data scientist who needs to process massive amounts of measurement data collected from gas turbines. He needs to generate a report containing the maintenance condition of hundreds of turbines daily. Therefore, his team needs a computing infrastructure to analyze the newly arrived data once a day. Batch jobs are run on a schedule and store aggregated results in a database. A business intelligence (BI) tool is used to generate reports based on the data stored in the database.

As the budget for computing infrastructure is very small, Nick and his team have been looking for a cost effective solution to analyze their data. He finds a way to make clever use of AWS's price model:

- *AWS bills virtual machines per minute.* So Nick launches a virtual machine when starting a batch job, and terminates it immediately after the job finished. Doing so allows him to pay for computing infrastructure only when actually using it. This is a big game changer compared to the traditional data center where Nick had to pay a monthly fee for each machine, no matter how much it was used.
- *AWS offers spare capacity in their data centers at substantial discount.* It is not important for Nick to run a batch job at a specific time. He can wait to execute a batch job until there is enough spare capacity available, so AWS offers him a virtual machine with a discount of 50%.

Figure 1.6 illustrates how Nick benefits from the pay-per-use price model for virtual machines.

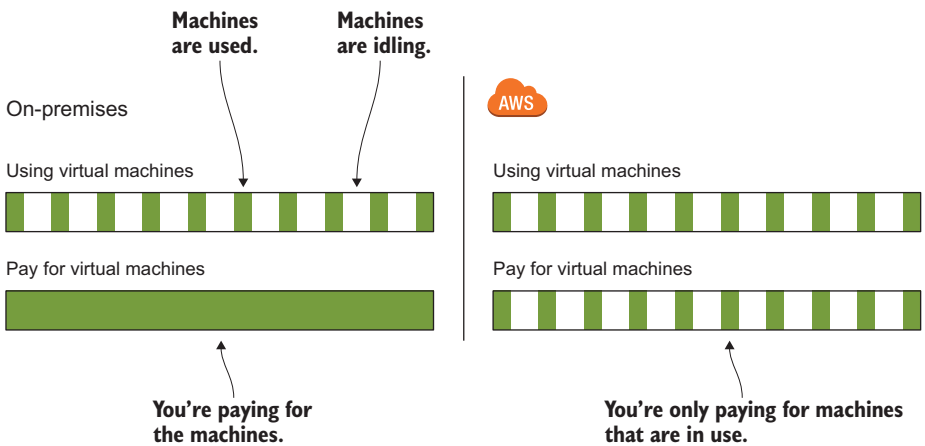


Figure 1.6 Making use of the pay-per-use price model of virtual machines

Nick is happy to have access to a computing infrastructure that allows his team to analyze data at low costs. You now have a broad idea of what you can do with AWS. Generally speaking, you can host any application on AWS. The next section explains the nine most important benefits AWS has to offer.

1.3 How you can benefit from using AWS

What's the most important advantage of using AWS? Cost savings, you might say. But saving money isn't the only advantage. Let's look at how else you can benefit from using AWS.

1.3.1 Innovative and fast-growing platform

AWS is announcing new services, features, and improvements constantly. Go to https://aws.amazon.com/about-aws/whats-new/ to get an impression of the speed of innovation. We have counted 719 announcements from Jan. 1 to Oct. 21 in 2017, and 641 announcements in 2016. Making use of the innovative technologies provided by AWS helps you to generate valuable solutions for your customers and thus achieve a competitive advantage.

AWS reported net sales of $4.1 billion USD for the quarter ending in June 2017. That's a year-over-year growth rate of 42% (Q3 2016 versus Q3 2017). We expect AWS to expand the size and extend of its platform in the upcoming years, for example, by adding additional services and data centers.[4]

1.3.2 Services solve common problems

As you've learned, AWS is a platform of services. Common problems such as load balancing, queuing, sending email, and storing files are solved for you by services. You don't need to reinvent the wheel. It's your job to pick the right services to build complex systems. So let AWS manage those services while you focus on your customers.

1.3.3 Enabling automation

Because AWS has an API, you can automate everything: you can write code to create networks, start virtual machine clusters, or deploy a relational database. Automation increases reliability and improves efficiency.

The more dependencies your system has, the more complex it gets. A human can quickly lose perspective, whereas a computer can cope with graphs of any size. You should concentrate on tasks humans are good at—such as describing a system—while the computer figures out how to resolve all those dependencies to create the system. Setting up an environment in the cloud based on your blueprints can be automated with the help of infrastructure as code, covered in chapter 4.

---

[4] Amazon, 10-Q for Quarter Ended June 30 (2017), http://mng.bz/1LAX.

1.3.4 Flexible capacity (scalability)

Flexible capacity frees you from planning. You can scale from one virtual machine to thousands of virtual machines. Your storage can grow from gigabytes to petabytes. You no longer need to predict your future capacity needs for the coming months and years.

If you run a web shop, you have seasonal traffic patterns, as shown in figure 1.7. Think about day versus night, and weekday versus weekend or holiday. Wouldn't it be nice if you could add capacity when traffic grows and remove capacity when traffic shrinks? That's exactly what flexible capacity is about. You can start new virtual machines within minutes and throw them away a few hours after that.

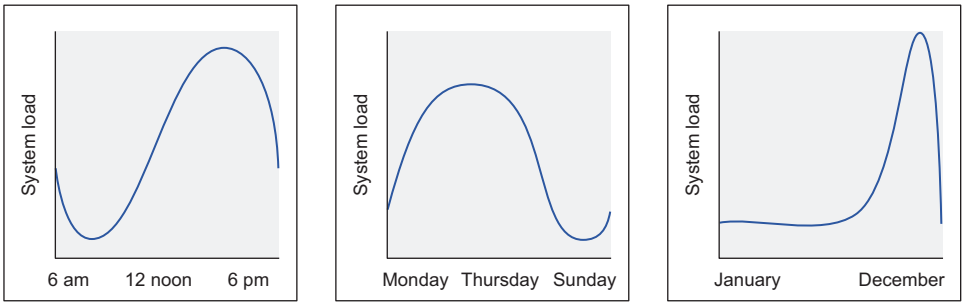


Figure 1.7 Seasonal traffic patterns for a web shop

The cloud has almost no capacity constraints. You no longer need to think about rack space, switches, and power supplies—you can add as many virtual machines as you like. If your data volume grows, you can always add new storage capacity.

Flexible capacity also means you can shut down unused systems. In one of our last projects, the test environment only ran from 7 a.m. to 8 p.m. on weekdays, allowing us to save 60%.

1.3.5 Built for failure (reliability)

Most AWS services are highly available or fault tolerant by default. If you use those services, you get reliability for free. AWS supports you as you build systems in a reliable way. It provides everything you need to create your own highly available or fault-tolerant systems.

1.3.6 Reducing time to market

In AWS, you request a new virtual machine, and a few minutes later that virtual machine is booted and ready to use. The same is true with any other AWS service available. You can use them all on demand.

Your development process will be faster because of the shorter feedback loops. You can eliminate constraints such as the number of test environments available; if you need another test environment, you can create it for a few hours.

1.3.7 Benefiting from economies of scale

AWS is increasing its global infrastructure constantly. Thus AWS benefits from an economy of scale. As a customer, you will benefit partially from these effects.

AWS reduces prices for their cloud services every now and then. A few examples:

- In November 2016, charges for storing data on the object storage S3 were reduced by 16% to 28%.
- In May 2017, prices were reduced by 10% to 17% for virtual machines with a one- or three-year commitment (reserved instances).
- In July 2017, AWS reduced prices for virtual machines running a Microsoft SQL Server (Standard Edition) by up to 52%.

1.3.8 Global infrastructure

Are you serving customers worldwide? Making use of AWS's global infrastructure has the following advantages: low network latencies between your customers and your infrastructure, being able to comply with regional data protection requirements, and benefiting from different infrastructure prices in different regions. AWS offers data centers in North America, South America, Europe, Asia, and Australia, so you can deploy your applications worldwide with little extra effort.

1.3.9 Professional partner

When you use AWS services, you can be sure that their quality and security follow the latest standards and certifications. For example:

- *ISO 27001*—A worldwide information security standard certified by an independent and accredited certification body.
- *ISO 9001*—A standardized quality management approach used worldwide and certified by an independent and accredited certification body.
- *PCI DSS Level 1*—A data security standard (DSS) for the payment card industry (PCI) to protect cardholders data.

Go to https://aws.amazon.com/compliance/ if you want to dive into the details. If you're still not convinced that AWS is a professional partner, you should know that Expedia, Vodafone, FDA, FINRA, Airbnb, Slack, and many more are running serious workloads on AWS.[5]

We have discussed a lot of reasons to run your workloads on AWS. But what does AWS cost? You will learn more about the pricing models in the next section.

1.4 How much does it cost?

A bill from AWS is similar to an electric bill. Services are billed based on use. You pay for the time a virtual machine was running, the used storage from the object store, or the number of running load balancers. Services are invoiced on a monthly basis. The

[5] AWS Customer Success, https://aws.amazon.com/solutions/case-studies/.

pricing for each service is publicly available; if you want to calculate the monthly cost of a planned setup, you can use the AWS Simple Monthly Calculator (http://aws.amazon.com/calculator).

1.4.1 Free Tier

You can use some AWS services for free within the first 12 months of your signing up. The idea behind the Free Tier is to enable you to experiment with AWS and get some experience using its services. Here is a taste of what's included in the Free Tier:

- 750 hours (roughly a month) of a small virtual machine running Linux or Windows. This means you can run one virtual machine for a whole month or you can run 750 virtual machines for one hour.
- 750 hours (or roughly a month) of a classic or application load balancer.
- Object store with 5 GB of storage.
- Small database with 20 GB of storage, including backup.

If you exceed the limits of the Free Tier, you start paying for the resources you consume without further notice. You'll receive a bill at the end of the month. We'll show you how to monitor your costs before you begin using AWS.

After your one-year trial period ends, you pay for all resources you use. But some resources are free forever. For example, the first 25 GB of the NoSQL database are free forever.

You get additional benefits, as detailed at http://aws.amazon.com/free. This book will use the Free Tier as much as possible and will clearly state when additional resources are required that aren't covered by the Free Tier.

1.4.2 Billing example

As mentioned earlier, you can be billed in several ways:

- *Based on minutes or hours of usage*—A virtual machine is billed per minute. A load balancer is billed per hour.
- *Based on traffic*—Traffic is measured in gigabytes or in number of requests, for example.
- *Based on storage usage*—Usage can be measured by capacity (for example, 50 GB volume no matter how much you use) or real usage (such as 2.3 GB used).

Remember the web shop example from section 1.2? Figure 1.8 shows the web shop and adds information about how each part is billed.

Let's assume your web shop started successfully in January, and you ran a marketing campaign to increase sales for the next month. Lucky you: you were able to increase the number of visitors to your web shop fivefold in February. As you already know, you have to pay for AWS based on usage. Table 1.1 shows your bill for February. The number of visitors increased from 100,000 to 500,000, and your monthly bill increased from $127 USD to $495 USD, which is a 3.9-fold increase. Because your web shop had to handle more traffic, you had to pay more for services, such as the CDN,

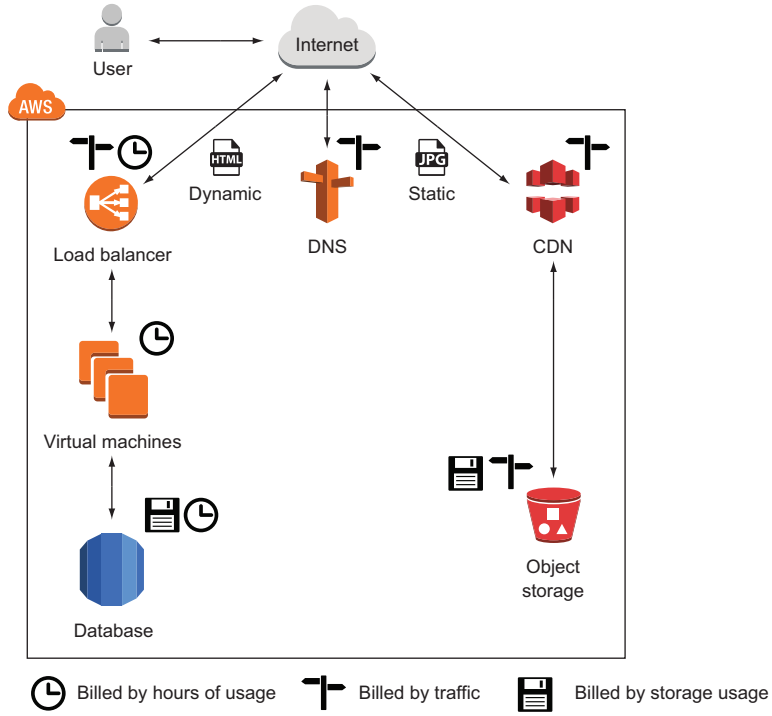


Figure 1.8 AWS bills services on minutes or hours of usage, by traffic, or by used storage.

the web servers, and the database. Other services, like the amount of storage needed for static files, didn't change, so the price stayed the same.

Table 1.1 How an AWS bill changes if the number of web shop visitors increases

Service	January usage	February usage	February charge	Increase
Visits to website	100,000	500,000		
CDN	25 M requests + 25 GB traffic	125 M requests + 125 GB traffic	$135.63 USD	$107.50 USD
Static files	50 GB used storage	50 GB used storage	$1.15 USD	$0.00 USD
Load balancer	748 hours + 50 GB traffic	748 hours + 250 GB traffic	$20.70 USD	$1.60 USD
Web servers	1 virtual machine = 748 hours	4 virtual machines = 2,992 hours	$200.46 USD	$150.35 USD
Database (748 hours)	Small virtual machine + 20 GB storage	Large virtual machine + 20 GB storage	$133.20 USD	$105.47 USD
DNS	2 M requests	10 M requests	$4.00 USD	$3.20 USD
Total cost			***$495.14 USD***	***$368.12 USD***

With AWS, you can achieve a linear relationship between traffic and costs. And other opportunities await you with this pricing model.

1.4.3 Pay-per-use opportunities

The AWS pay-per-use pricing model creates new opportunities. For example, the barrier for starting a new project is lowered, as you no longer need to invest in infrastructure up front. You can start virtual machines on demand and only pay per second of usage, and you can stop using those virtual machines whenever you like and no longer have to pay for them. You don't need to make an upfront commitment regarding how much storage you'll use.

Another example: a big server costs exactly as much as two smaller ones with the same capacity. Thus you can divide your systems into smaller parts, because the cost is the same. This makes fault tolerance affordable not only for big companies but also for smaller budgets.

1.5 Comparing alternatives

AWS isn't the only cloud computing provider. Microsoft Azure and Google Cloud Platform (GCP) are major players as well.

The three major cloud providers share a lot in common. They all have:

- A worldwide infrastructure that provides computing, networking, and storage capabilities.
- An IaaS offering that provides virtual machines on-demand: Amazon EC2, Azure Virtual Machines, Google Compute Engine.
- Highly distributed storage systems able to scale storage and I/O capacity without limits: Amazon S3, Azure Blob storage, Google Cloud Storage.
- A pay-as-you-go pricing model.

But what are the differences between the cloud providers?

AWS is the market leader in cloud computing, offering an extensive product portfolio. Even if AWS has expanded into the enterprise sector during recent years, it is still obvious that AWS started with services to solve internet-scale problems. Overall, AWS is building great services based on innovative, mostly open source, technologies. AWS offers complicated but rock-solid ways to restrict access to your cloud infrastructure.

Microsoft Azure provides Microsoft's technology stack in the cloud, recently expanding into web-centric and open source technologies as well. It seems like Microsoft is putting a lot of effort into catching up with Amazon's market share in cloud computing.

GCP is focused on developers looking to build sophisticated distributed systems. Google combines their worldwide infrastructure to offer scalable and fault-tolerant services (such as Google Cloud Load Balancing). The GCP seems more focused on cloud-native applications than on migrating your locally hosted applications to the cloud, in our opinion.

There are no shortcuts to making an informed decision about which cloud provider to choose. Each use case and project is different. The devil is in the details. Also don't forget where you are coming from. (Are you using Microsoft technology heavily? Do you have a big team consisting of system administrators or are you a developer-centric company?) Overall, in our opinion, AWS is the most mature and powerful cloud platform available at the moment.

1.6 Exploring AWS services

Hardware for computing, storing, and networking is the foundation of the AWS cloud. AWS runs services on this hardware, as shown in figure 1.9. The API acts as an interface between AWS services and your applications.

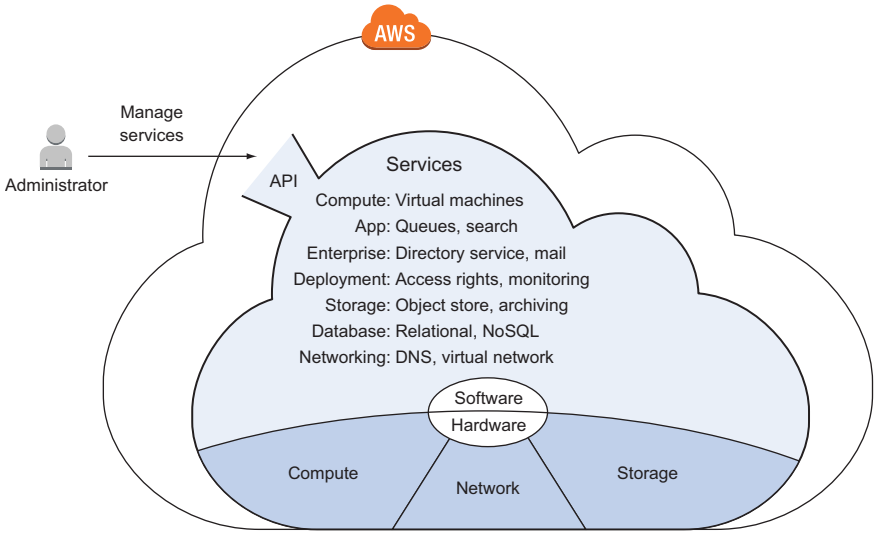


Figure 1.9 The AWS cloud is composed of hardware and software services accessible via an API.

You can manage services by sending requests to the API manually via a web-based UI like the Management Console, a command-line interface (CLI), or programmatically via an SDK. Virtual machines have a special feature: you can connect to virtual machines through SSH, for example, and gain administrator access. This means you can install any software you like on a virtual machine. Other services, like the NoSQL database service, offer their features through an API and hide everything that's going on behind the scenes. Figure 1.10 shows an administrator installing a custom PHP web application on a virtual machine and managing dependent services such as a NoSQL database used by the application.

Users send HTTP requests to a virtual machine. This virtual machine is running a web server along with a custom PHP web application. The web application needs to talk to AWS services in order to answer HTTP requests from users. For example, the

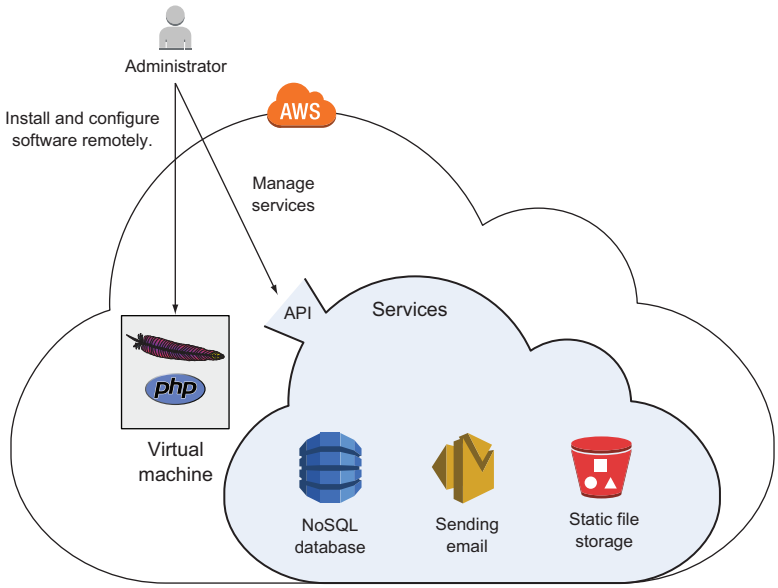


Figure 1.10 Managing a custom application running on a virtual machine and dependent services

application might need to query data from a NoSQL database, store static files, and send email. Communication between the web application and AWS services is handled by the API, as figure 1.11 shows.

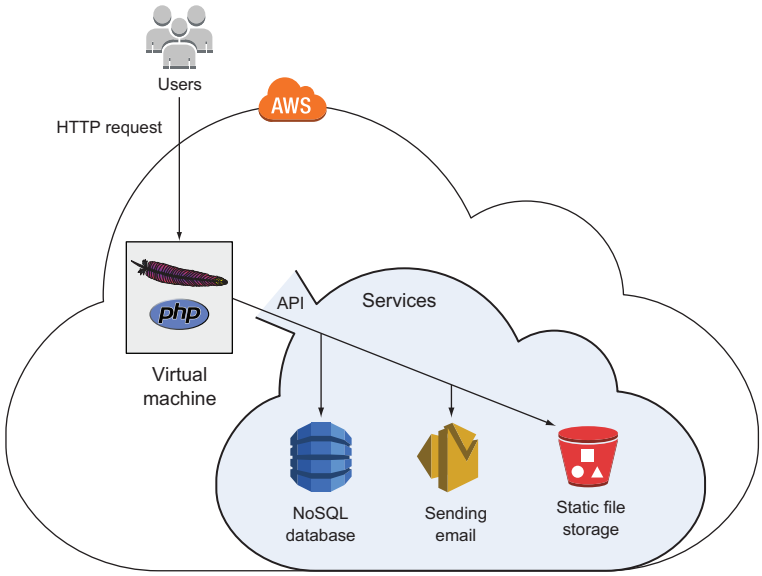


Figure 1.11 Handling an HTTP request with a custom web application using additional AWS services

The number of services available can be scary at the outset. When logging into AWS's web interface you are presented with an overview listing 98 services. On top of that, new services are announced constantly during the year and at the big conference in Las Vegas, AWS re:Invent.

AWS offers services in the following categories:

- Analytics
- Application Integration
- AR and VR
- Business Productivity
- Compute
- Customer Engagement
- Database
- Desktop and App Streaming
- Developer Tools
- Game Development
- Internet Of Things
- Machine Learning
- Management Tools
- Media Services
- Migration
- Mobile Services
- Networking and Content Delivery
- Security, Identity, and Compliance
- Storage

Unfortunately, it is not possible to cover all services offered by AWS in our book. Therefore, we are focusing on the services that will best help you get started quickly, as well as the most widely used services. The following services are covered in detail in our book:

- *EC2*—Virtual machines
- *ELB*—Load balancers
- *Lambda*—Executing functions
- *Elastic Beanstalk*—Deploying web applications
- *S3*—Object store
- *EFS*—Network filesystem
- *Glacier*—Archiving data
- *RDS*—SQL databases
- *DynamoDB*—NoSQL database
- *ElastiCache*—In-memory key-value store
- *VPC*—Private network
- *CloudWatch*—Monitoring and logging
- *CloudFormation*—Automating your infrastructure
- *OpsWorks*—Deploying web applications
- *IAM*—Restricting access to your cloud resources
- *Simple Queue Service*—Distributed queues

We are missing at least three important topics that would fill their own books: continuous delivery, Docker/containers, and Big Data. Let us know when you are interested in reading one of these unwritten books.

But how do you interact with an AWS service? The next section explains how to use the web interface, the CLI, and SDKs to manage and access AWS resources.

1.7 Interacting with AWS

When you interact with AWS to configure or use services, you make calls to the API. The API is the entry point to AWS, as figure 1.12 demonstrates.

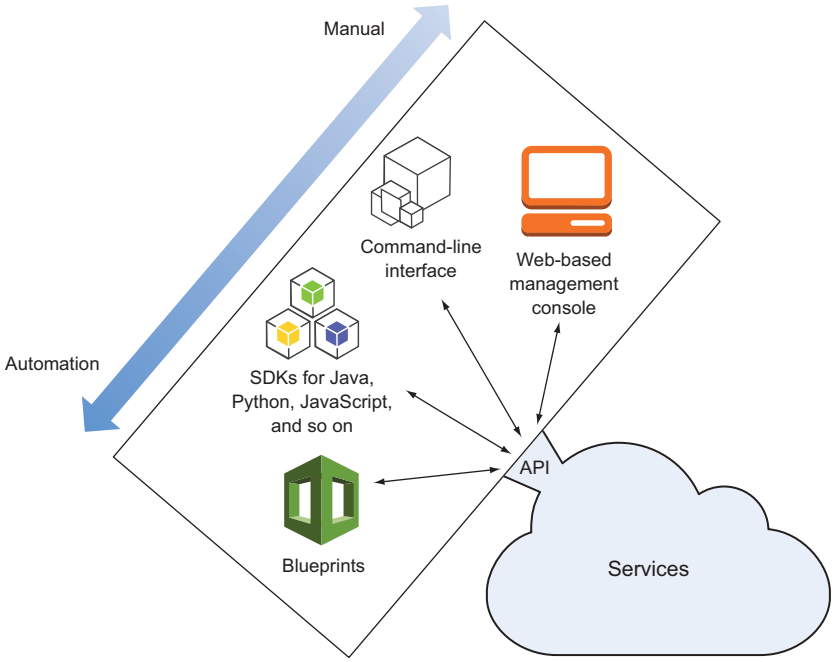


Figure 1.12 Different ways to access the AWS API, allowing you to manage and access AWS services

Next, we'll give you an overview of the tools available for communicating with API: the Management Console, the command-line interface, the SDKs, and infrastructure blueprints. We will compare the different tools, and you will learn how to use all of them while working your way through the book.

1.7.1 Management Console

The AWS Management Console allows you to manage and access AWS services through a graphical user interface (GUI), which runs in every modern web browser (the latest three versions of Google Chrome and Mozilla Firefox; Apple Safari: 9, 8, and 7; Microsoft Internet Explorer: 11; Microsoft Edge: 12). See figure 1.13.

When getting started or experimenting with AWS, the Management Console is the best place to start. It helps you to gain an overview of the different services quickly. The Management Console is also a good way to set up a cloud infrastructure for development and testing.

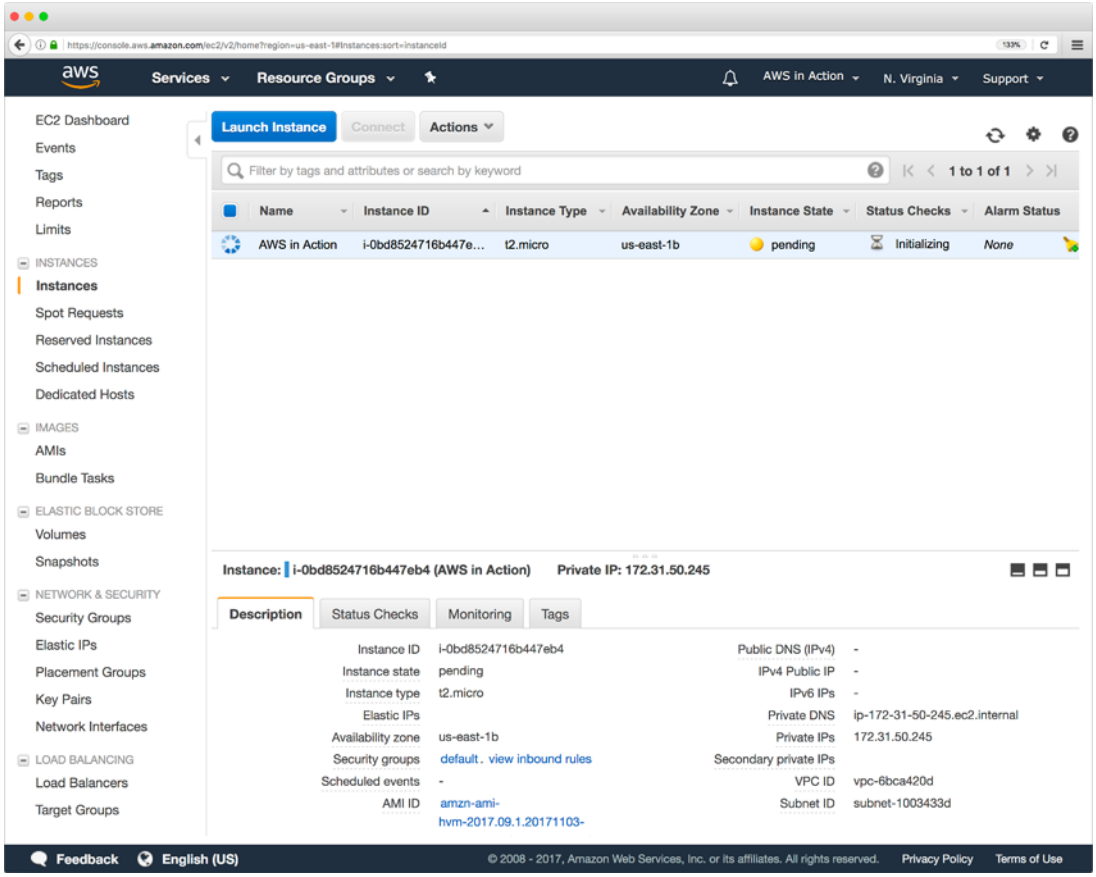


Figure 1.13 The AWS Management Console offers a GUI to manage and access AWS services.

1.7.2 Command-line interface

The command-line interface (CLI) allows you to manage and access AWS services within your terminal. Because you can use your terminal to automate or semi-automate recurring tasks, CLI is a valuable tool. You can use the terminal to create new cloud infrastructures based on blueprints, upload files to the object store, or get the details of your infrastructure's networking configuration regularly. Figure 1.14 shows the CLI in action.

If you want to automate parts of your infrastructure with the help of a continuous integration server, like Jenkins, the CLI is the right tool for the job. The CLI offers a convenient way to access the API and combine multiple calls into a script.

You can even begin to automate your infrastructure with scripts by chaining multiple CLI calls together. The CLI is available for Windows, Mac, and Linux, and there is also a PowerShell version available.

Figure 1.14 The CLI allows you to manage and access AWS services from your terminal.

1.7.3 SDKs

Use your favorite programming language to interact with the AWS API. AWS offers SDKs for the following platforms and languages:

- Android
- Browsers (JavaScript)
- iOS
- Java
- .NET
- Node.js (JavaScript)
- PHP
- Python
- Ruby
- Go
- C++

SDKs are typically used to integrate AWS services into applications. If you're doing software development and want to integrate an AWS service like a NoSQL database or a push-notification service, an SDK is the right choice for the job. Some services, such as queues and topics, must be used with an SDK.

1.7.4 Blueprints

A *blueprint* is a description of your system containing all resources and their dependencies. An Infrastructure as Code tool compares your blueprint with the current system, and calculates the steps to create, update, or delete your cloud infrastructure. Figure 1.15 shows how a blueprint is transferred into a running system.

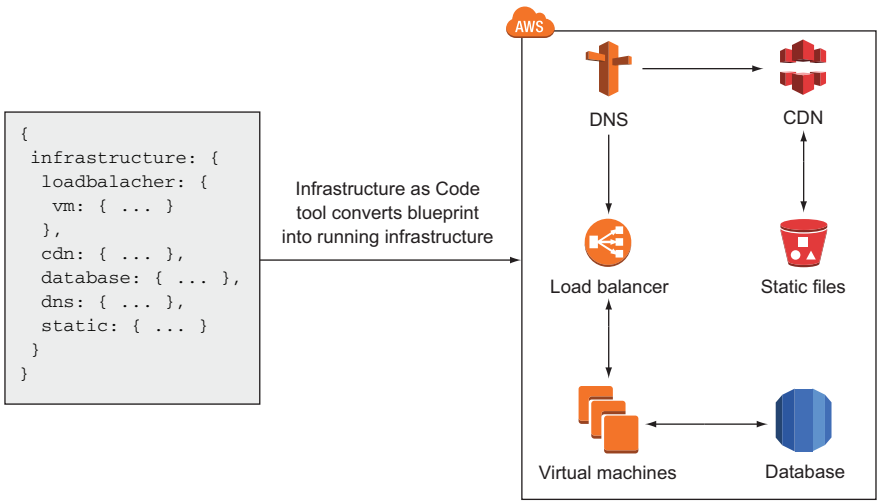


Figure 1.15 Infrastructure automation with blueprints

Consider using blueprints if you have to control many or complex environments. Blueprints will help you to automate the configuration of your infrastructure in the cloud. You can use them to set up a network and launch virtual machines, for example.

Automating your infrastructure is also possible by writing your own source code with the help of the CLI or the SDKs. But doing so requires you to resolve dependencies, make sure you are able to update different versions of your infrastructure, and handle errors yourself. As you will see in chapter 4, using a blueprint and an Infrastructure-as-Code tool solves these challenges for you. It's time to get started creating your AWS account and exploring AWS practice after all that theory.

1.8 Creating an AWS account

Before you can start using AWS, you need to create an account. Your account is a basket for all your cloud resources. You can attach multiple users to an account if multiple humans need access to it; by default, your account will have one root user. To create an account, you need the following:

- A telephone number to validate your identity
- A credit card to pay your bills

USING AN OLD ACCOUNT? It is possible to use your existing AWS account while working through this book. In this case, your usage might not be covered by the Free Tier. So you might have to pay for the use.

If you created your existing AWS account before Dec. 4, 2013, please create a new one, as there are some legacy issues that might cause trouble during our examples.

1.8.1 Signing up

The sign-up process consists of five steps:

1. Provide your login credentials.
2. Provide your contact information.
3. Provide your payment details.
4. Verify your identity.
5. Choose your support plan.

Point your favorite web browser to https://aws.amazon.com, and click the *Create a Free Account* button.

1. Providing your login credentials

Creating an AWS account starts with defining a unique AWS account name, as shown in figure 1.16. The AWS account name has to be globally unique among all AWS customers. Try `aws-in-action-$yourname` and replace `$yourname` with your name. Beside the account name, you have to specify an email address and a password used to authenticate the root user of your AWS account.

We advise you to choose a strong password to prevent misuse of your account. *Use a password consisting of at least 20 characters.* Protecting your AWS account from unwanted access is crucial to avoid data breaches, data loss, or unwanted resource usage on your behalf.

Figure 1.16 Creating an AWS account: sign-up page

2. PROVIDING YOUR CONTACT INFORMATION

The next step, as shown in figure 1.17, is adding your contact information. Fill in all the required fields, and continue.

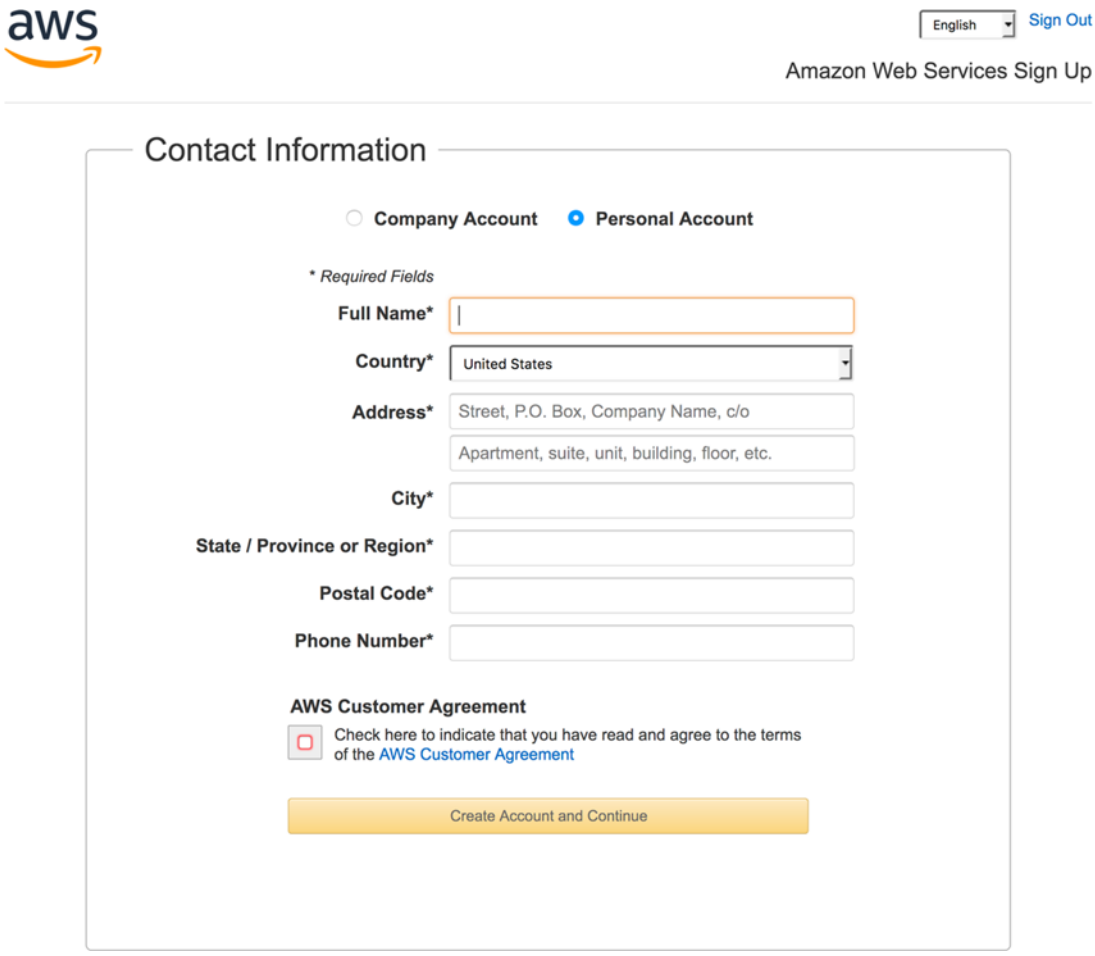


Figure 1.17 Creating an AWS account: providing your contact information

3. PROVIDING YOUR PAYMENT DETAILS

Next the screen shown in figure 1.18 asks for your payment information. Provide your credit card information. There's an option to change the currency setting from USD to AUD, CAD, CHF, DKK, EUR, GBP, HKD, JPY, NOK, NZD, SEK, or ZAR later on if that's more convenient for you. If you choose this option, the amount in USD is converted into your local currency at the end of the month.

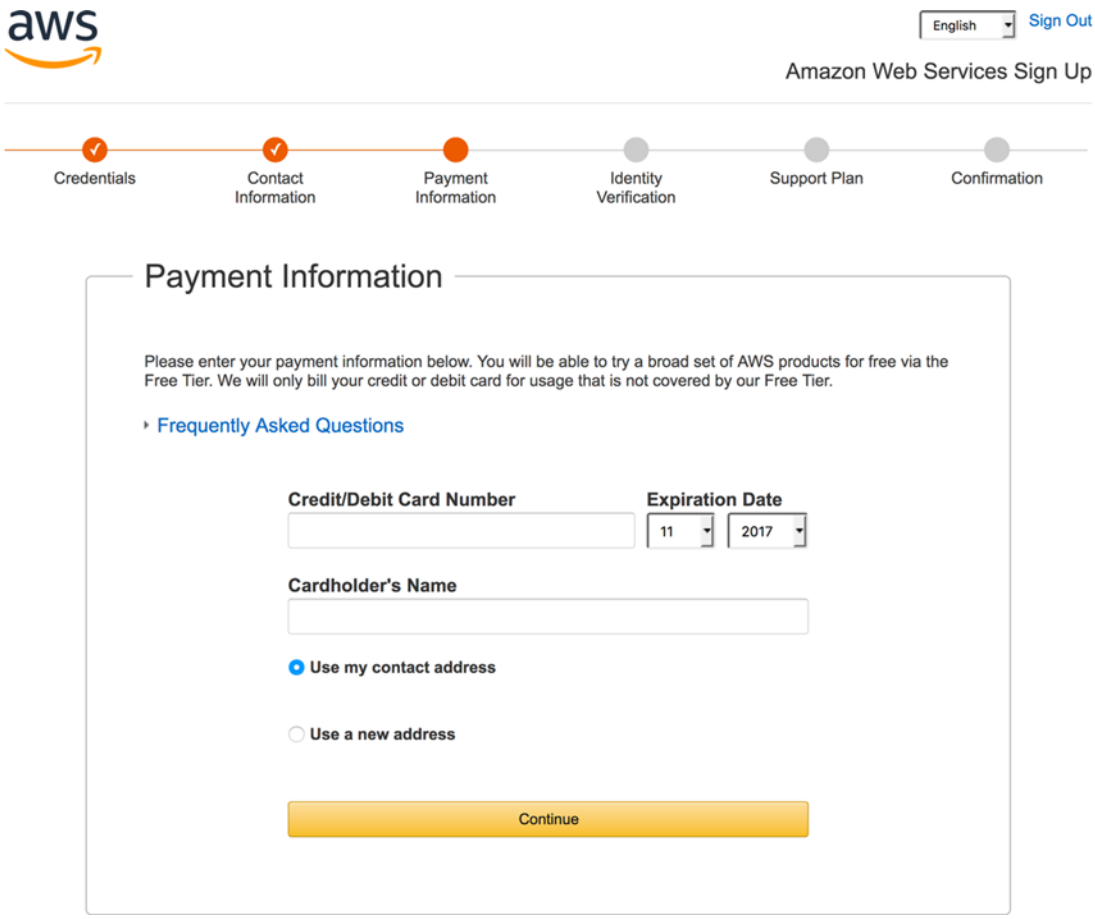


Figure 1.18 Creating an AWS account: providing your payment details

4. Verifying your identity

The next step is to verify your identity. Figure 1.19 shows the first step of the process. After you complete the first part of the form, you'll receive a call from AWS. A robot voice will ask for your PIN. The four-digit PIN is displayed on the website and you have to enter it using your telephone. After your identity has been verified, you are ready to continue with the last step.

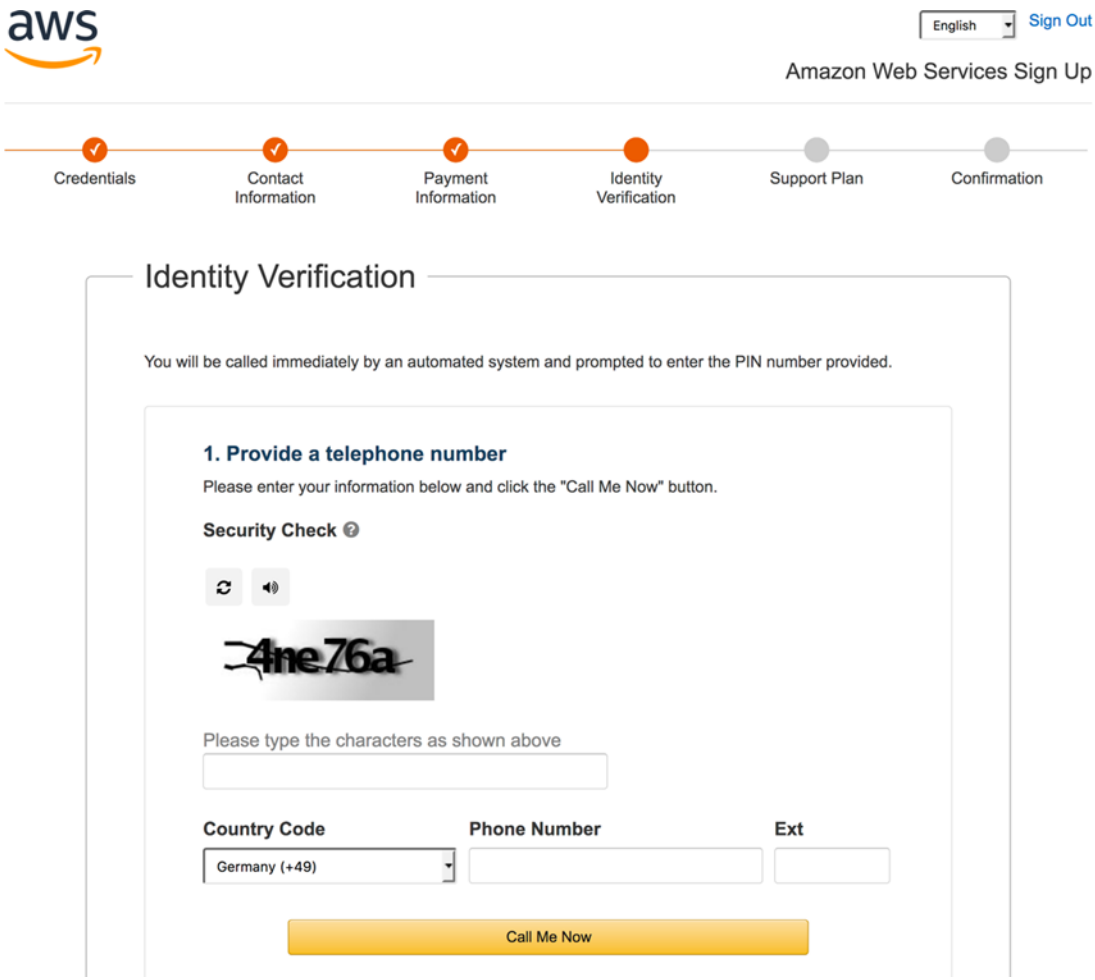


Figure 1.19 Creating an AWS account: verifying your identity

5. CHOOSING YOUR SUPPORT PLAN

The last step is to choose a support plan; see figure 1.20. In this case, select the Basic plan, which is free. If you later create an AWS account for your business, we recommend the Business support plan. You can even switch support plans later.

High five! You're done. Click Launch Management Console as shown in figure 1.21 to sign into your AWS account for the first time.

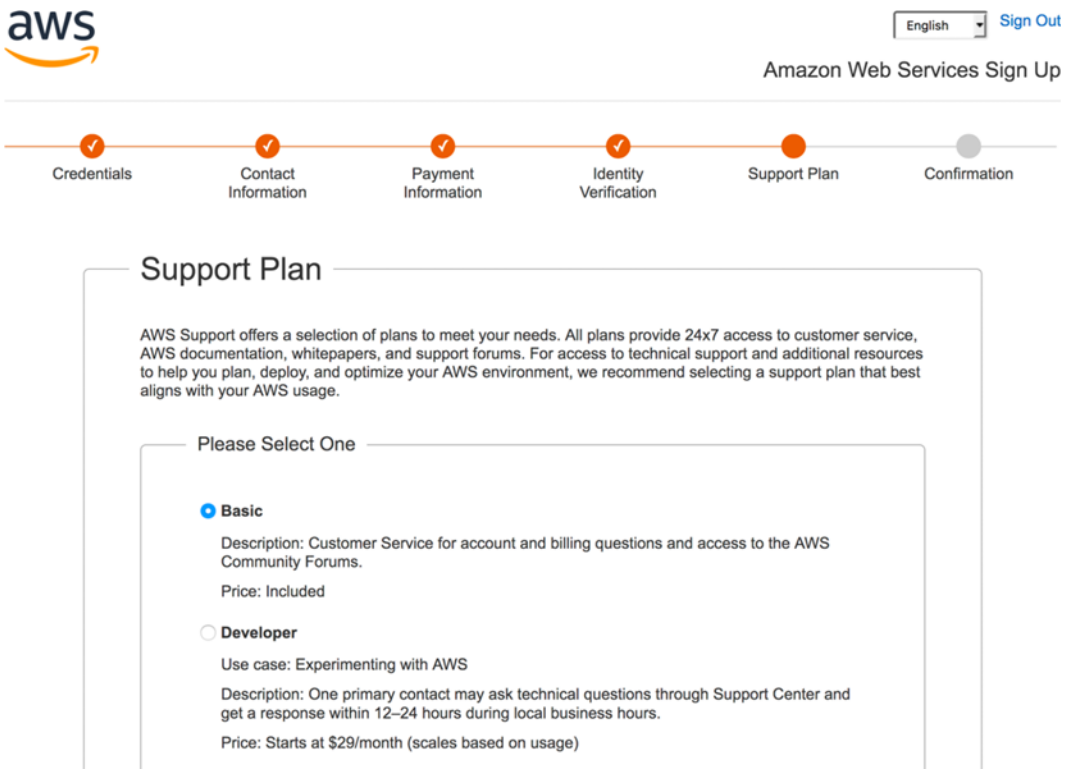


Figure 1.20 Creating an AWS account: choosing your support plan

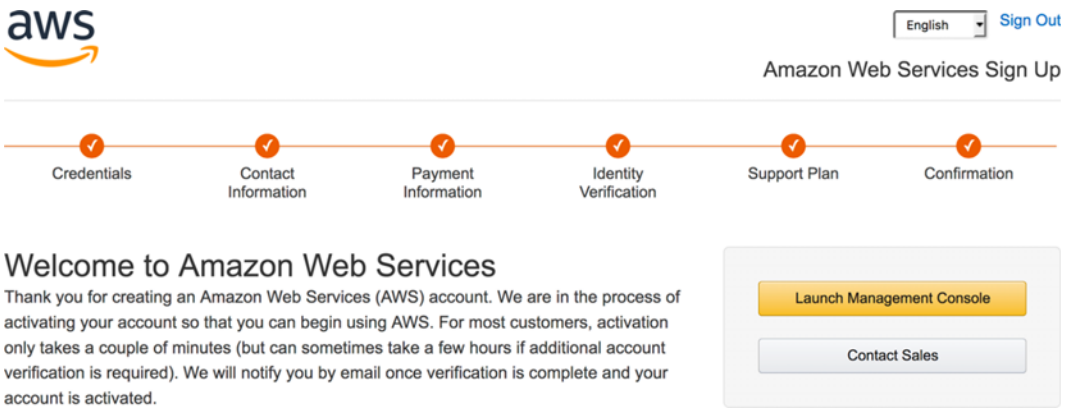


Figure 1.21 You have successfully created an AWS account.

1.8.2 Signing In

You now have an AWS account and are ready to sign in to the AWS Management Console. As mentioned earlier, the Management Console is a web-based tool you can use to control AWS resources; it makes most of the functionality of the AWS API available to you. Figure 1.22 shows the sign-in form at https://console.aws.amazon.com. Enter your email address, click Next, and then enter your password to sign in.

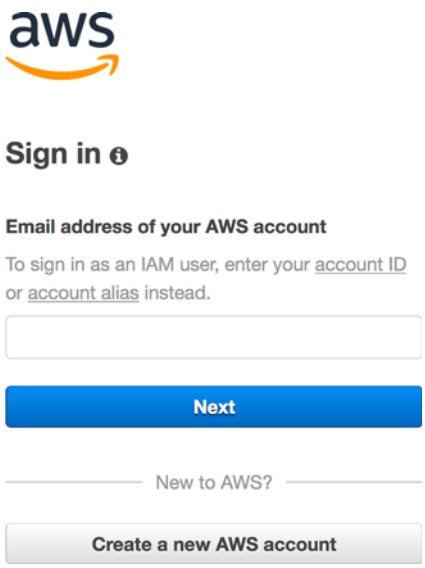


Figure 1.22 Sign in to the Management Console.

After you have signed in successfully, you are forwarded to the start page of the Management Console, as shown in figure 1.23.

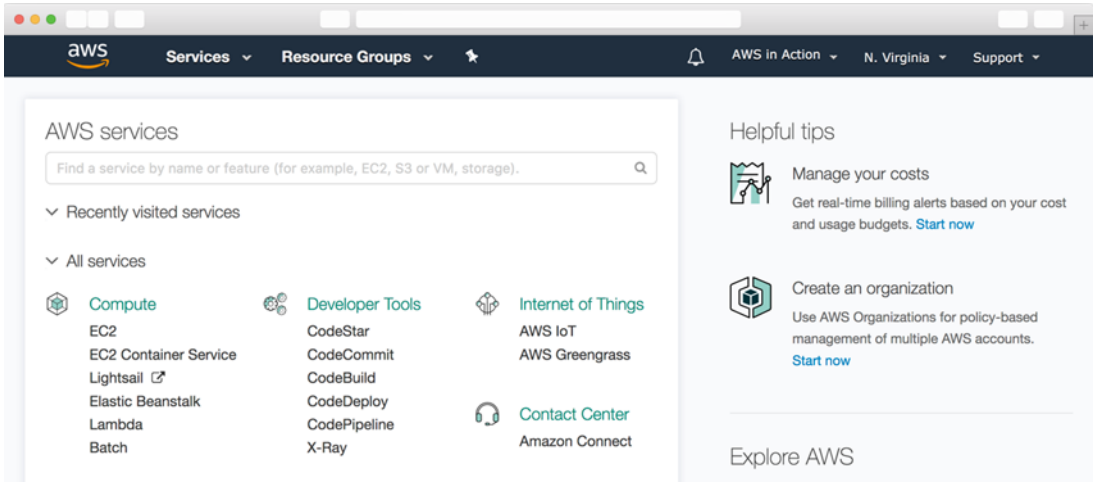


Figure 1.23 AWS Management Console

The most important part is the navigation bar at the top, shown in figure 1.24. It consists of seven sections:

- *AWS*—Start page of the Management Console, including an overview of all services.
- *Services*—Provides quick access to all AWS services.
- *Resource Groups*—Allows you to get an overview of all your AWS resources.
- *Custom section (Edit)*—Click the edit icon and drag-and-drop important services here to personalize the navigation bar.
- *Your name*—Lets you access billing information and your account, and also lets you sign out.
- *Your region*—Lets you choose your region. You will get to know about regions in section 3.5. You don't need to change anything here right now.
- *Support*—Gives you access to forums, documentation, and a ticket system.

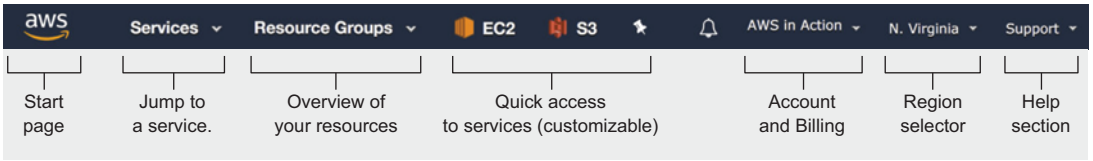


Figure 1.24 AWS Management Console navigation bar

Next, you'll create a key pair so you can connect to your virtual machines.

1.8.3 Creating a key pair

A *key pair* consists of a private key and a public key. The public key will be uploaded to AWS and injected into the virtual machine. The private key is yours; it's like your password, but much more secure. Protect your private key as if it were a password. It's your secret, so don't lose it—you can't retrieve it.

To access a Linux machine, you use the SSH protocol; you'll use a key pair for authentication instead of a password during login. When accessing a Windows machine via Remote Desktop Protocol (RDP), you'll need a key pair to decrypt the administrator password before you can log in.

> **Region US East (N. Virginia)**
>
> Amazon operates data centers in various geographic regions around the world. To simplify the examples, we're using the region US East (N. Virginia) within our book. You will also learn how to switch to another region to use resources in Asia Pacific (Sydney).
>
> Make sure you have selected the region US East (N. Virginia) before creating your key pair. Use the region selector in the navigation bar of the Management Console to change the region if needed.

The following steps will guide you to the dashboard of the EC2 service, which offers virtual machines, and where you can obtain a key pair:

1. Open the AWS Management Console at https://console.aws.amazon.com.
2. Click Services in the navigation bar and select EC2.
3. Your browser should now show the EC2 Dashboard.

The EC2 Dashboard, shown in figure 1.25, is split into three columns. The first column is the EC2 navigation bar; because EC2 is one of the oldest services, it has many features that you can access via the navigation bar. The second column gives you a brief overview of all your EC2 resources. The third column provides additional information.

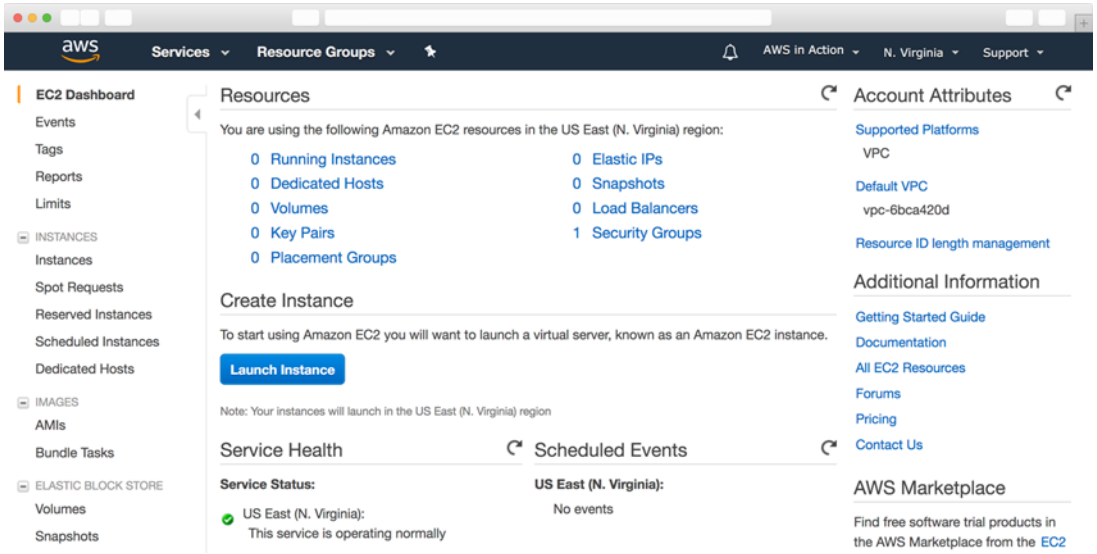


Figure 1.25 EC2 Management Console

Follow the steps in figure 1.26 to create a new key pair:

1. Click Key Pairs in the navigation bar under Network & Security.
2. Click the Create Key Pair button.
3. Name the Key Pair `mykey`. If you choose another name, you must replace the name in all the following examples during the whole book!

During the key-pair creation process, you download a file called mykey.pem. You must now prepare that key for future use. Depending on your operating system, you need to do things differently, so please read the section that fits your OS.

Figures with cueballs

In some figures, as in figure 1.26, you'll see numbered cueballs. They mark the order in which you need to click to follow the process being discussed in the text.

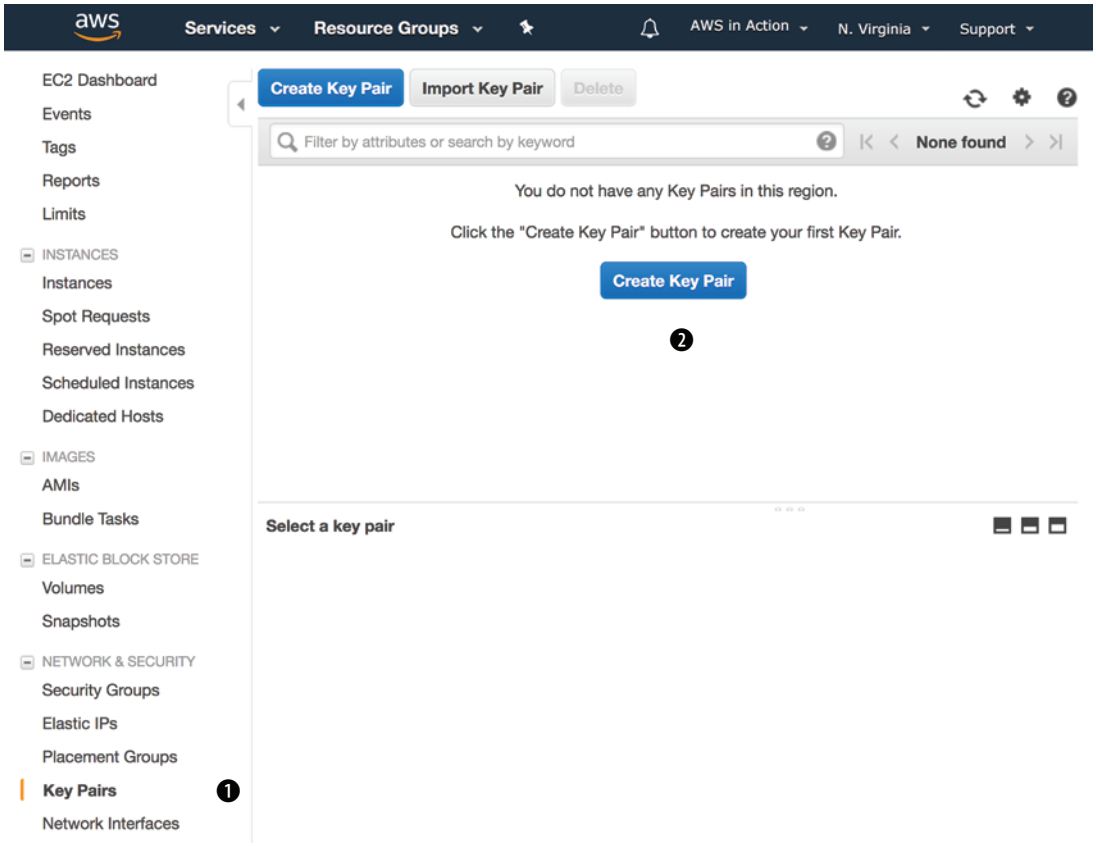


Figure 1.26 EC2 Management Console Key Pairs

Using your own key pair

It's also possible to upload the public key part from an existing key pair to AWS. Doing so has two advantages:

- You can reuse an existing key pair.
- You can be sure only you know the private key part of the key pair. If you use the Create Key Pair button, AWS knows (at least briefly) your private key.

We decided against that approach in this case because it's less convenient to implement in a book.

LINUX AND MACOS

The only thing you need to do is change the access rights of mykey.pem so that only you can read the file. To do so, run `chmod 400 mykey.pem` in the terminal. You'll learn about how to use your key when you need to log in to a virtual machine for the first time in this book.

WINDOWS

Windows doesn't ship an SSH client, so you need to download the PuTTY installer for Windows from http://mng.bz/A1bY and install PuTTY. PuTTY comes with a tool called PuTTYgen that can convert the mykey.pem file into a mykey.ppk file, which you'll need:

1. Run the PuTTYgen application. The screen shown in figure 1.27 opens. The most important steps are highlighted on the screen.
2. Select RSA (or SSH-2 RSA) under Type of Key to Generate.
3. Click Load.
4. Because PuTTYgen displays only *.ppk files, you need to switch the file extension of the File Name field to All Files.
5. Select the mykey.pem file, and click Open.
6. Confirm the dialog box.
7. Change Key Comment to `mykey`.
8. Click Save Private Key. Ignore the warning about saving the key without a passphrase.

Your .pem file has now been converted to the .ppk format needed by PuTTY. You'll learn about how to use your key when you need to log in to a virtual machine for the first time in this book.

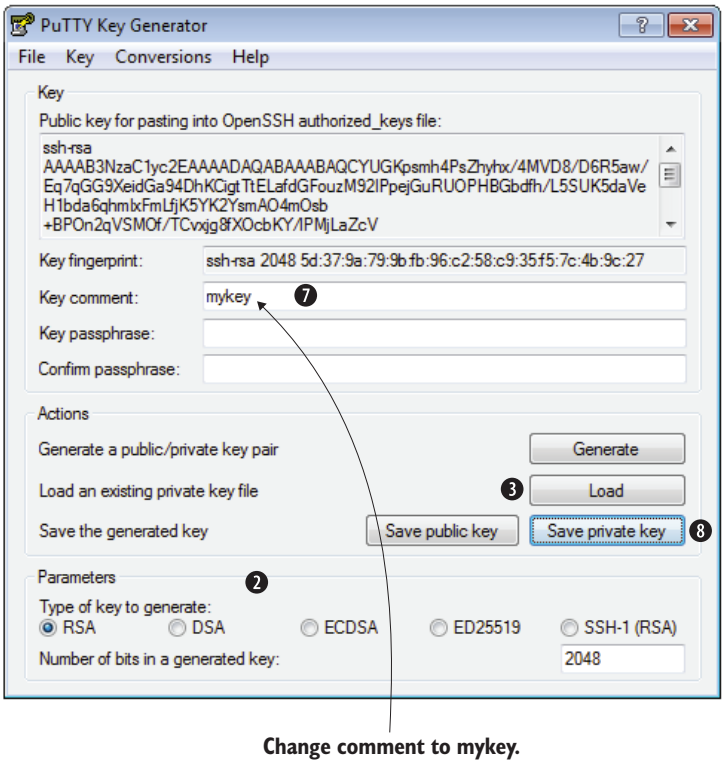


Figure 1.27 PuTTYgen allows you to convert the downloaded .pem file into the .ppk file format needed by PuTTY.

1.9 Create a billing alarm to keep track of your AWS bill

At first, the pay-per-use pricing model of AWS might feel unfamiliar to you, as it is not 100% foreseeable what your bill will look like at the end of the month. Most of the examples in this book are covered by the Free Tier, so AWS won't charge you anything. Exceptions are clearly marked. To provide you with the peace of mind needed to learn about AWS in a comfortable environment, you will create a billing alarm next. The billing alarm will notify you via email if your monthly AWS bill exceeds $5 USD so that you can react quickly.

First, you need to enable billing alarms within your AWS account. The step are illustrated in figure 1.28. The first step, of course, is to open the AWS Management Console at https://console.aws.amazon.com.

1. Click your Name in the main navigation bar on the top.
2. Select My Billing Dashboard from the pop-up menu.
3. Go to Preferences by using the sub navigation on the left side.
4. Select the Receive Billing Alerts check box.
5. Click Save preferences.

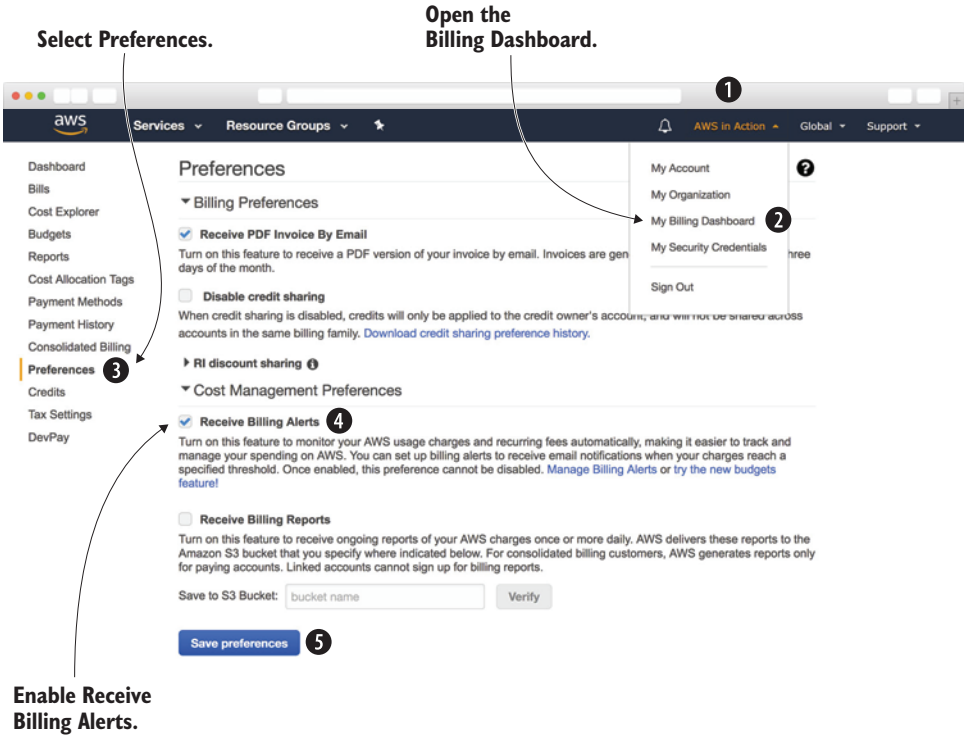


Figure 1.28 Creating a billing alarm (step 1 of 4)

You are now able to create a billing alarm. Here are the steps to do so:

1. Open the AWS Management Console at https://console.aws.amazon.com.
2. Open Services in the navigation bar and select CloudWatch.
3. Click the Create a billing alarm link as shown in figure 1.29.

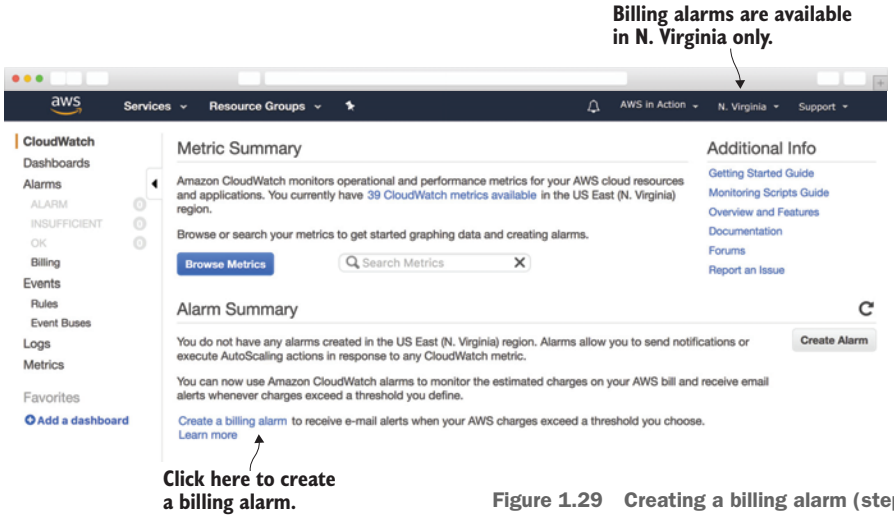


Figure 1.29 Creating a billing alarm (step 2 of 4)

Figure 1.30 shows the wizard that guides you through creating a billing alarm. Enter the threshold of total monthly charges for the billing alarm. We suggest $5 UDS, the equivalent of the price for a cup of coffee, as the threshold. Type in the email address where you want to receive notifications from your billing alarm in case your AWS bill exceeds the threshold. Click Create Alarm to create your billing alarm.

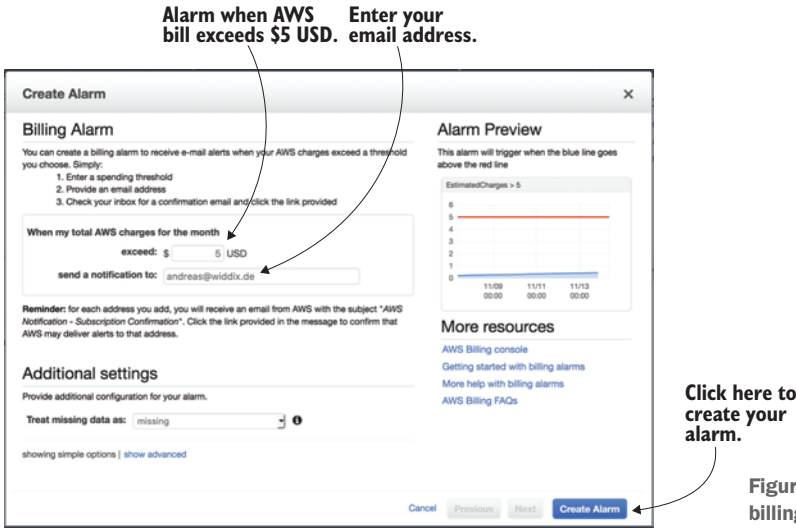


Figure 1.30 Creating a billing alarm (step 3 of 4)

Figure 1.31 Creating a billing alarm (step 4 of 4)

Open your inbox. An email from AWS containing a confirmation link appears. Click the confirmation link to complete the setup of your billing alarm. The dialog shown in figure 1.31 shows the status.

That's all. If your monthly AWS bill exceeds $5 USD for any reason, you will be notified immediately, allowing you to react before unwanted costs occur.

Summary

- Amazon Web Services (AWS) is a platform of web services for computing, storing, and networking that work well together.
- Cost savings aren't the only benefit of using AWS. You'll also profit from an innovative and fast-growing platform with flexible capacity, fault-tolerant services, and a worldwide infrastructure.
- Any use case can be implemented on AWS, whether it's a widely used web application or a specialized enterprise application with an advanced networking setup.
- You can interact with AWS in many different ways. You can control the different services by using the web-based GUI, use code to manage AWS programmatically from the command line or SDKs, or use blueprints to set up, modify, or delete your infrastructure on AWS.
- Pay-per-use is the pricing model for AWS services. Computing power, storage, and networking services are billed similarly to electricity.
- Creating an AWS account is easy. Now you know how to set up a key pair so you can log in to virtual machines for later use.
- Creating a billing alarm allows you to keep track of your AWS bill and get notified whenever you exceed the Free Tier.

CLOUD/AWS

Amazon Web Services IN ACTION Second Edition

Michael Wittig • Andreas Wittig

The largest and most mature of the cloud platforms, AWS offers over 100 prebuilt services, practically limitless compute resources, bottomless secure storage, as well as top-notch automation capabilities. This book shows you how to develop, host, and manage applications on AWS.

Amazon Web Services in Action, Second Edition is a comprehensive introduction to deploying web applications in the AWS cloud. You'll find clear, relevant coverage of all essential AWS services, with a focus on automation, security, high availability, and scalability. This thoroughly revised edition covers the latest additions to AWS, including serverless infrastructure with AWS Lambda, sharing data with EFS, and in-memory storage with ElastiCache.

What's Inside

- Completely revised bestseller!
- Secure and scale distributed applications
- Deploy applications on AWS
- Design for failure to achieve high availability
- Automate your infrastructure

Written for mid-level developers and DevOps engineers.

Andreas and **Michael Wittig** are software engineers and DevOps consultants focused on AWS. Together, they migrated the first bank in Germany to AWS in 2013.

To download their free eBook in PDF, ePub, and Kindle formats, owners of this book should visit manning.com/books/amazon-web-services-in-action-second-edition

"Slices through the complexity of AWS using examples and visuals to cement knowledge in the minds of readers."

—From the Foreword by Ben Whaley
AWS community hero and author

"The authors' ability to explain complex concepts is the real strength of the book."

—Antonio Pessolano
Consoft Sistemi

"Useful examples, figures, and sources to help you learn efficiently."

—Christof Marte, Daimler-Benz

"Does a great job of explaining some of the key services in plain English so you have the knowledge necessary to dig deeper."

—Ryan Burrows
Rooster Park Consulting

ISBN-13: 978-1-61729-511-9
ISBN-10: 1-61729-511-6
55499
9 781617 295119

MANNING $54.99 / Can $72.99 [INCLUDING eBOOK]