

COMP5349– Cloud Computing

Week 1: Intro to Cloud Computing

Dr. Ying Zhou

The University of Sydney

Table of Contents

COMMONWEALTH OF AUSTRALIA

Copyright Regulations 1969

WARNING

This material has been reproduced and communicated to you by or on behalf of the **University of Sydney** pursuant to Part VB of the Copyright Act 1968 (the Act).

The material in this communication may be subject to copyright under the Act. Any further reproduction or communication of this material by you may be the subject of copyright protection under the Act.

Do not remove this notice

- 01 What is Cloud Computing
- 02 Service Specification and Pricing
- 03 Introduction to Amazon Web Services
- 04 AWS Sample Applications
- 05 Cost of AWS
- 06 Exploring AWS Services
- 07 Interacting with AWS Services
- 08 Accessing AWS Services

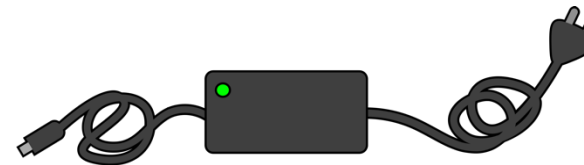
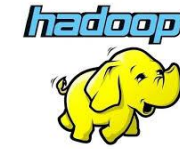
Some slides are adapted/adopted from AWS Academy Course

What is Cloud Computing

What is "Cloud"?

- Informally, we may view cloud computing as a way of *renting/sharing* IT resources
 - **Through Internet/Web**
 - Has an innovative way to specify, measure and charge the rented resources
 - Many other features...
- Not every kind of IT resources renting is called cloud
 - Lease from Dell to equip our labs
 - Rent some space from your ISP to set up a website

What are IT resources?



To run a renting business

- A way to package/measure/quantify your rental product/service
- A way to charge the customer
 - Hourly/daily/monthly/yearly rate
 - Subscription
- A way to deliver the produce/service
 - Truck, courier, pickup, or **Internet**
- A way to guarantee your product/service meet the client's requirements
- The particular form "Cloud" comes after supporting technologies are mature, and of course, good incentives for providers and market needs.

Cloud Computing– a Broad Definition

- A definition by US Governments' National Institute of Standard and Technology
 - “Cloud computing is a model for enabling ubiquitous, convenient, on-demand **network access** to a **shared pool of configurable computing resources** (e.g., *networks, servers, storage, applications, and services*)”
- In early days, we tend to differentiate three different models
 - Infrastructure as a Service (IaaS)
 - Platform as a Service (PaaS)
 - Software as a Service (SaaS)
- There are many new services and
 - Many providers are not restricted by a single service model
 - Many services cannot be categorized easily

Typical architecture in a cloud environment

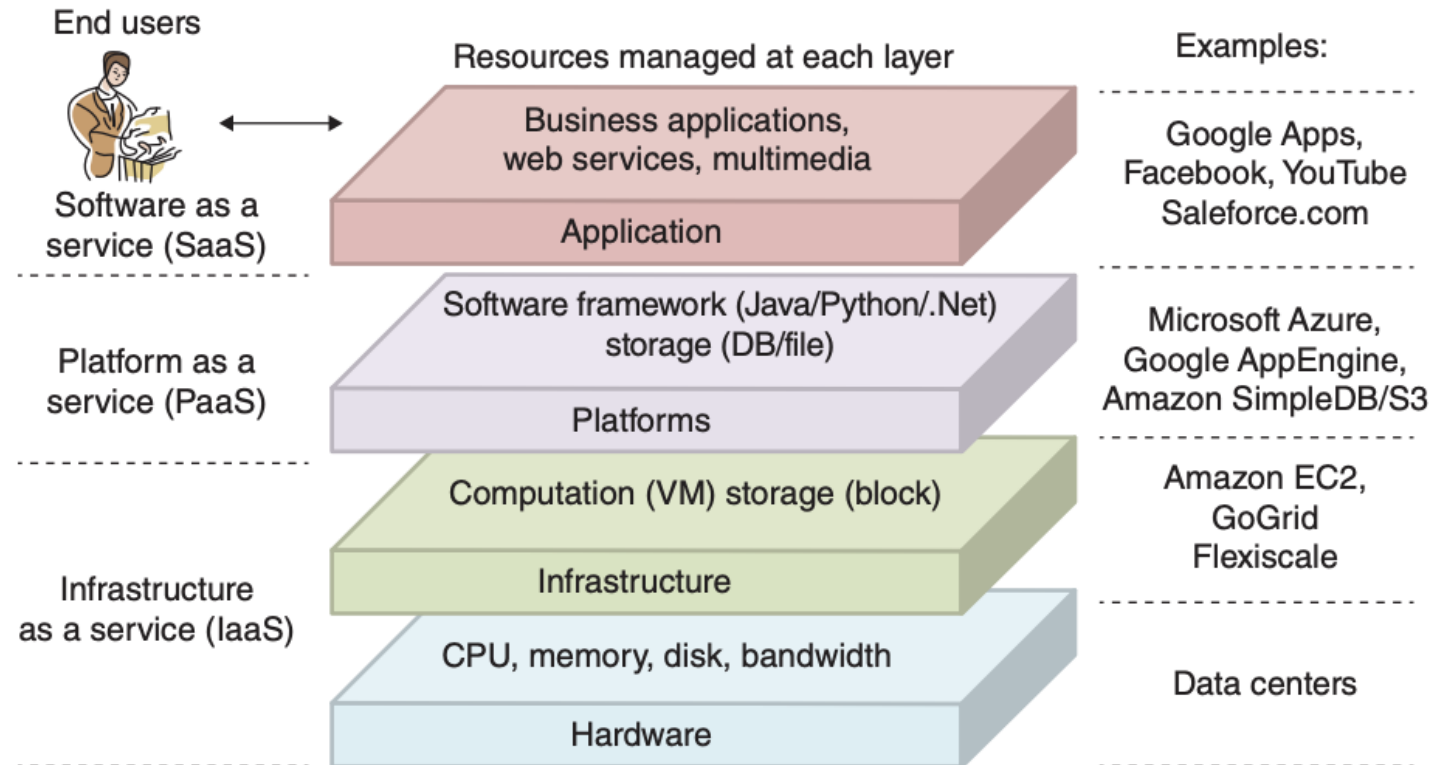


Figure 1.1. Typical architecture in a cloud computing environment.

SaaS Examples

- **Software as a Service (SaaS):** The consumer uses an **application**, but does not control the operating system, hardware or network infrastructure on which it's running.
 - Examples
 - Business applications: CRM solutions from *salesforce.com*
 - Business/Personal applications: Gmail, Google Doc, etc.
- SaaS in many ways are different to the other models



Gmail for business

25 GB storage, less spam and a 99.9% uptime SLA and enhanced email security.



Google Calendar

Agenda management, scheduling, shared online calendars and mobile calendar sync.



Google Docs

Documents, spreadsheets, drawings and presentations. Work online without attachments.



Service Cloud

PaaS Examples

- **Platform as a Service (PaaS):** The consumer uses a **hosting environment** for their applications. The consumer controls the applications that run in the environment (and possibly has some control over the hosting environment), but does not control the operating system, hardware or network infrastructure on which they are running. The platform is typically an **application framework**.

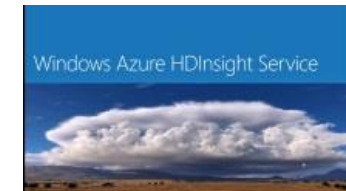
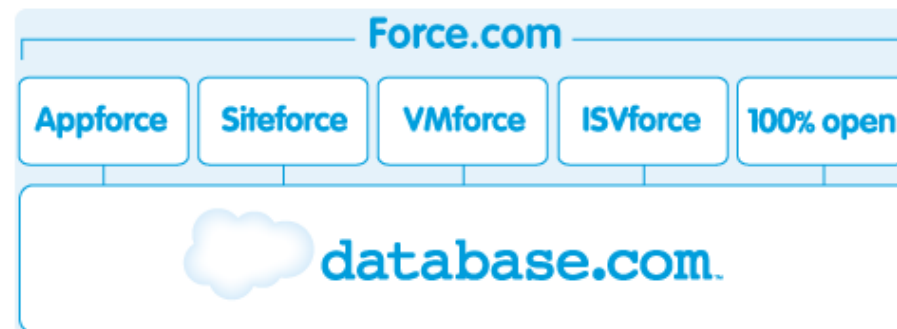
Google App Engine

Hor



Run your web apps on Google's infrastructure.

Easy to build, easy to maintain, easy to scale.

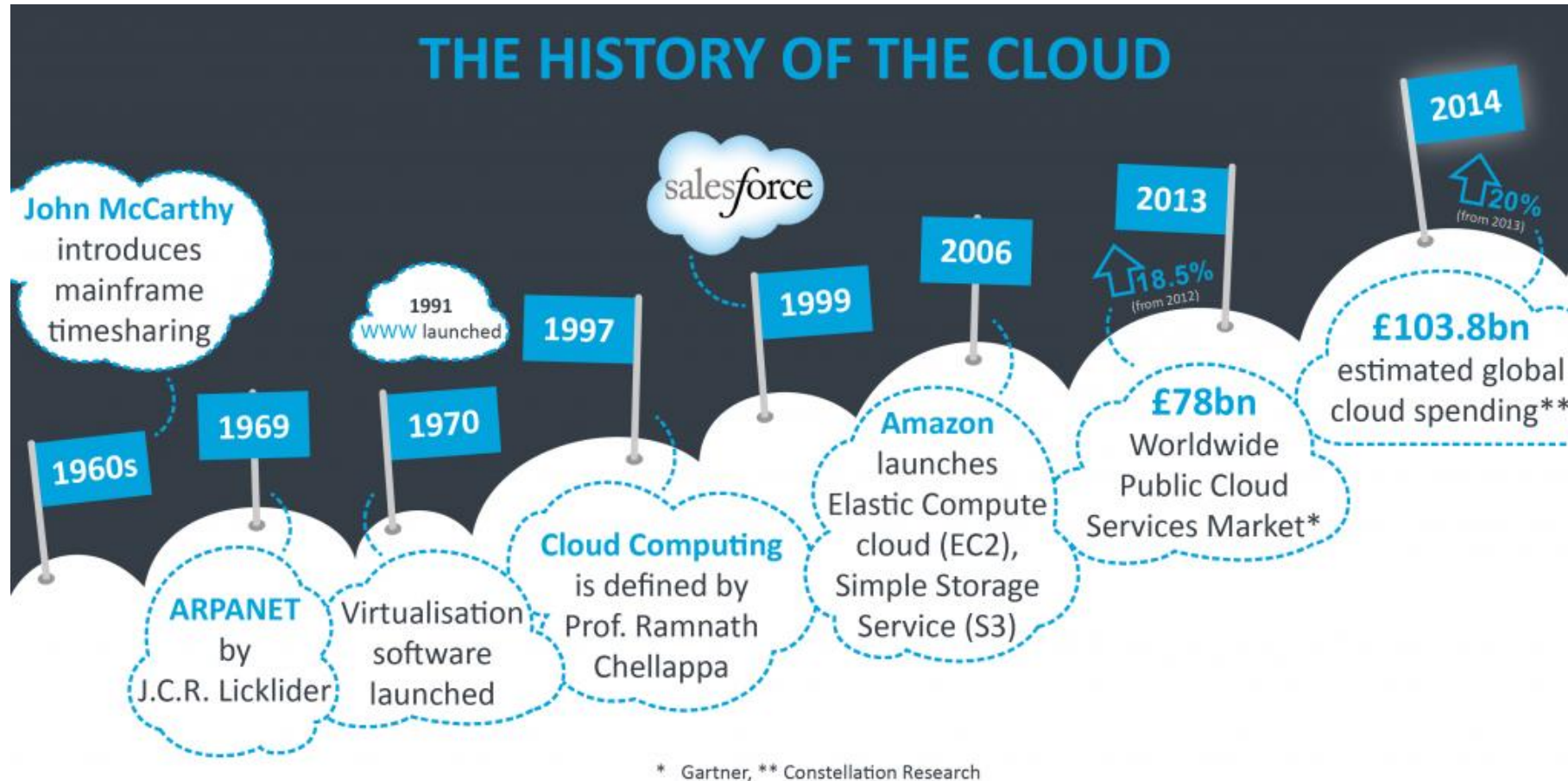


AWS Elastic MapReduce

IaaS Examples

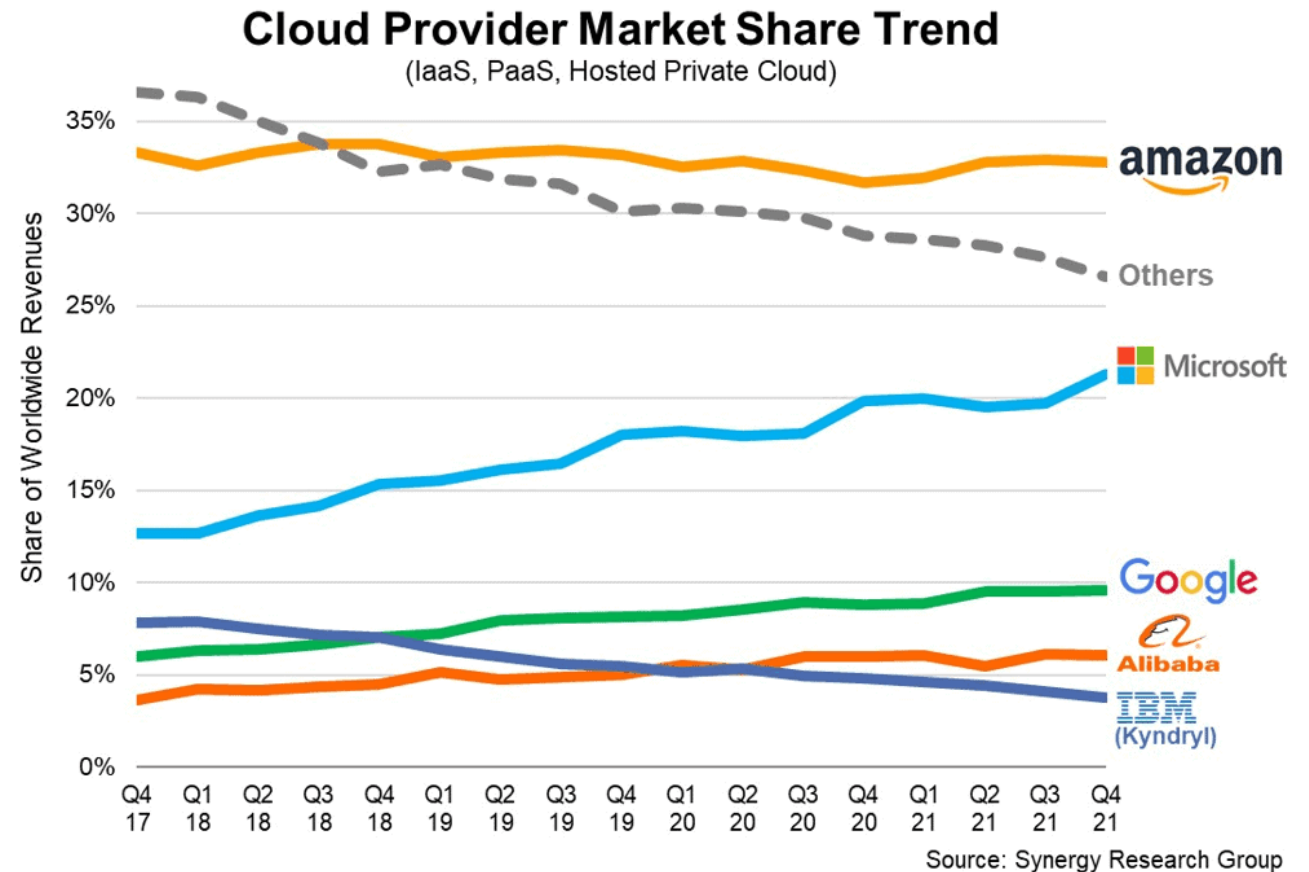
- › **Infrastructure as a Service (IaaS):** The consumer uses "**fundamental computing resources**" such as processing power, storage, networking components or middleware. The consumer can control the operating system, storage, deployed applications and possibly networking components such as firewalls and load balancers, but not the cloud infrastructure beneath them.





Major players

- Amazon
 - Amazon launched Amazon Web Services in 2006
- Microsoft
 - Microsoft Azure is officially released in 2010
- Google
 - Google App Engine was released in 2008 (an early PaaS service)
 - Google Cloud Platform was launched in 2011



Service Specification and Pricing

SaaS service specification and pricing

- SaaS
 - The service specification depends on the actual application, it could be the number of user account supported, the size of storage, etc
 - The pricing is usually subscription based, e.g. monthly or yearly price

<https://products.office.com/en-au/compare-all-microsoft-office-products?tab=2> accessed 07/03/2018

\$129.00

(per year)

Office 365 Home

Or buy for \$13.00 per month →

Best for households. Includes Office applications for up to 5 users.

Office applications included



Word



Excel



PowerPoint



OneNote



Outlook



Publisher
(PC only)



Access
(PC only)

Services included



OneDrive



Skype

IaaS Specification and Pricing

- IaaS
 - The specification is similar to the general spec when you purchase a computer. These include cpu speed, number of cores, memory, etc
 - At the beginning, most providers use fine grained pay-as-you-go hourly rate
 - Now many providers have even finer grained “Per Second Billing”[<https://aws.amazon.com/ec2/pricing/> accessed 07/03/2018]

Model	GPUs	vCPU	Mem (GiB)	GPU Mem (GiB)	GPU P2P
p3.2xlarge	1	8	61	16	-
p3.8xlarge	4	32	244	64	NVLink
p3.16xlarge	8	64	488	128	NVLink

- Up to 8 NVIDIA Tesla V100 GPUs, each pairing 5,120 CUDA Cores and 640 Tensor Cores
- High frequency Intel Xeon E5-2686 v4 (Broadwell) processors

p3.2xlarge	\$3.06 per Hour
p3.8xlarge	\$12.24 per Hour
p3.16xlarge	\$24.48 per Hour

NVIDIA Tesla V100 - GPU computing processor - Tesla V100 - 16 GB



Price:
\$8,720.99

1

Add to Cart



<https://aws.amazon.com/ec2/pricing/on-demand/>
<https://aws.amazon.com/ec2/instance-types/>
<http://www.zones.com/site/product/index.html?id=105374001>
accessed 07/03/2018

Specification and Pricing

- PaaS
 - Somewhere in between, could be fine grained hourly rate or subscription based.
 - E.g. If you start a MapReduce cluster in Azure or AWS, you can specify how many nodes you want to have and the node type, you will be charged hourly (or secondly) based on those instances' price.

App Engine applications run as instances within the [standard environment](#) or the [flexible environment](#).

Instances within the standard environment have access to a daily limit of resource usage that is provided at no charge defined by a set of [quotas](#). Beyond that level, applications will incur charges as outlined below. To control your application costs, you can set a [spending limit](#). To estimate costs for the standard environment, use the pricing calculator.

<https://cloud.google.com/appengine/pricing>

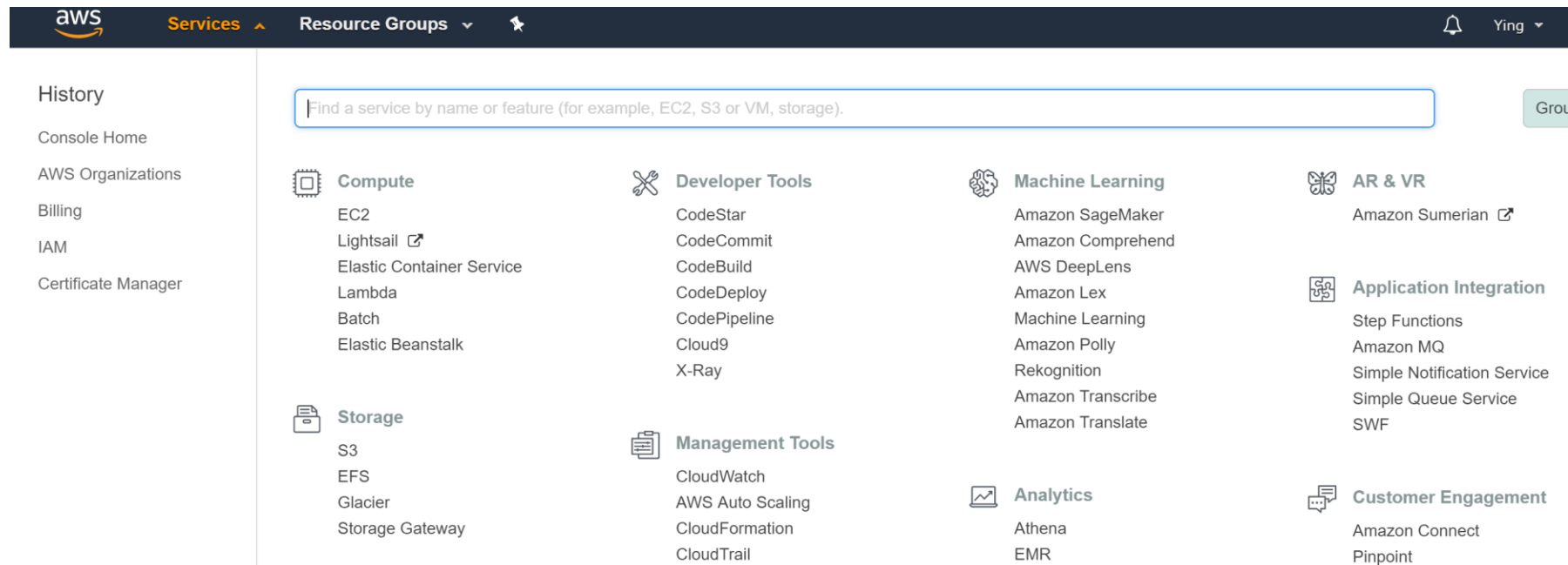
Sydney	
Instance class	Cost per hour per instance
B1	\$0.068
B2	\$0.135
B4	\$0.270

Other services

- Other services have their own way of describing, charging and enabling technologies
 - E.g. most storage services rely on company's own implementation of a planet scale storage system: Azure storage, DynamoDB, etc
 - Storage charging is more complicated as it has both the static and dynamic part
 - Actual storage size
 - Number of queries
 - Consistency and other quality requirement

Service Delivery

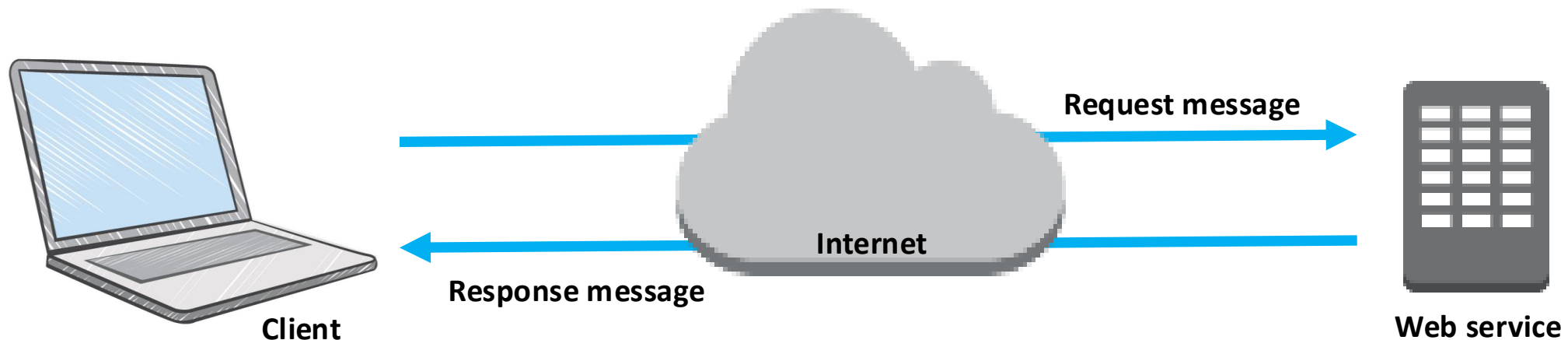
- All those XaaS models are delivered through Internet, with a web interface



Introduction to Amazon Web Services

What are web services?

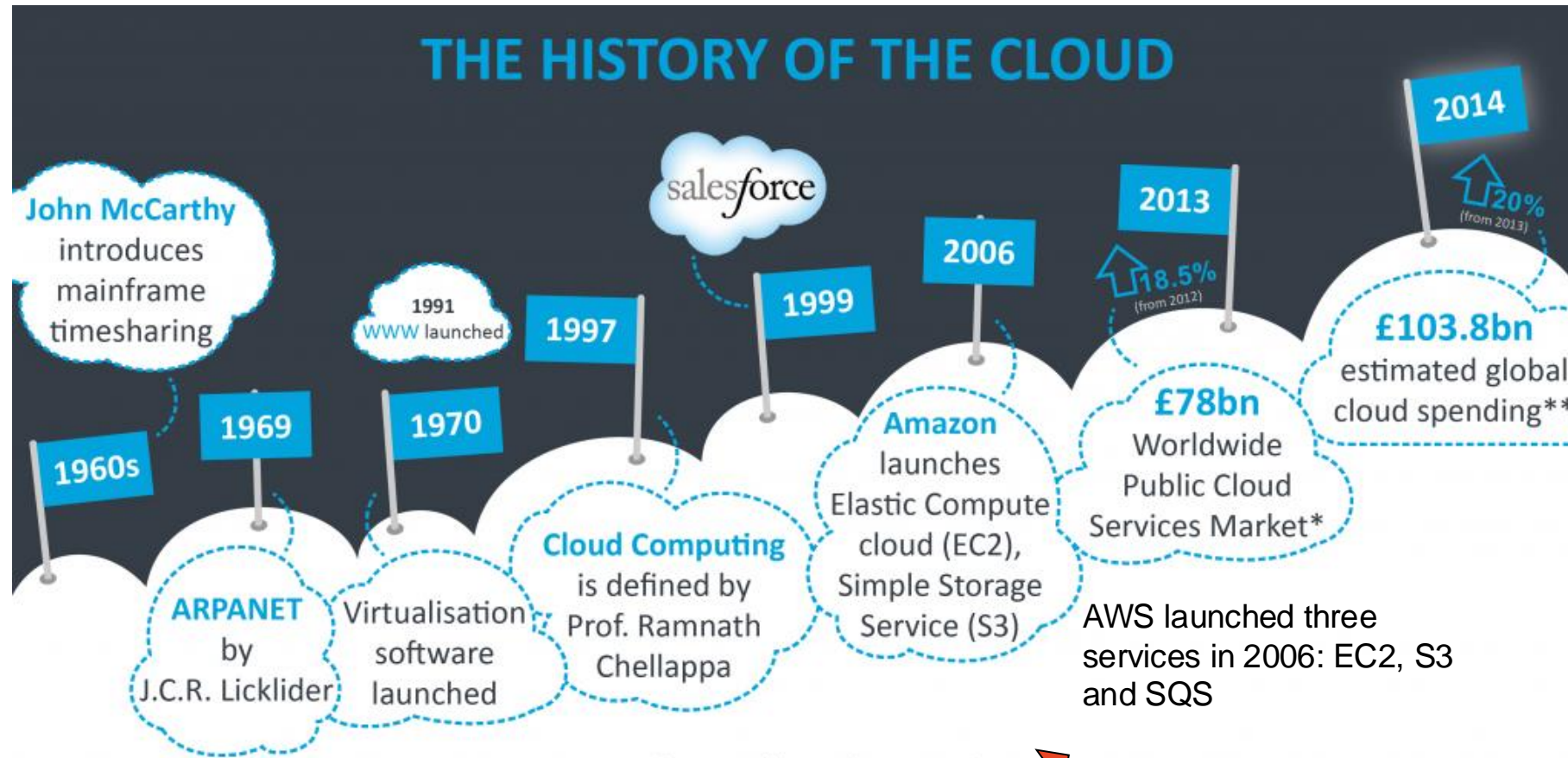
A **web service** is any piece of software that makes itself available over the internet and uses a **standardized format**—such as Extensible Markup Language (XML) or JavaScript Object Notation (JSON)—for the request and the response of an **application programming interface (API) interaction**.



What is AWS?

- Amazon Web Services (AWS) is a platform of web services that offers solutions for computing, storing, and networking, at different layers of abstraction.
- *Web services* are accessible via the internet by using typical web protocols (such as HTTP) and are used by machines or by humans through a UI.

The Web Services History of Amazon Cloud



* Gartner, ** Constellation Research

XML based SOAP web services was proposed in 1998



Amazon published a few SOAP bases services in 2002



The physical part of AWS

- The services are supported from data centers across the world.



* Limited access

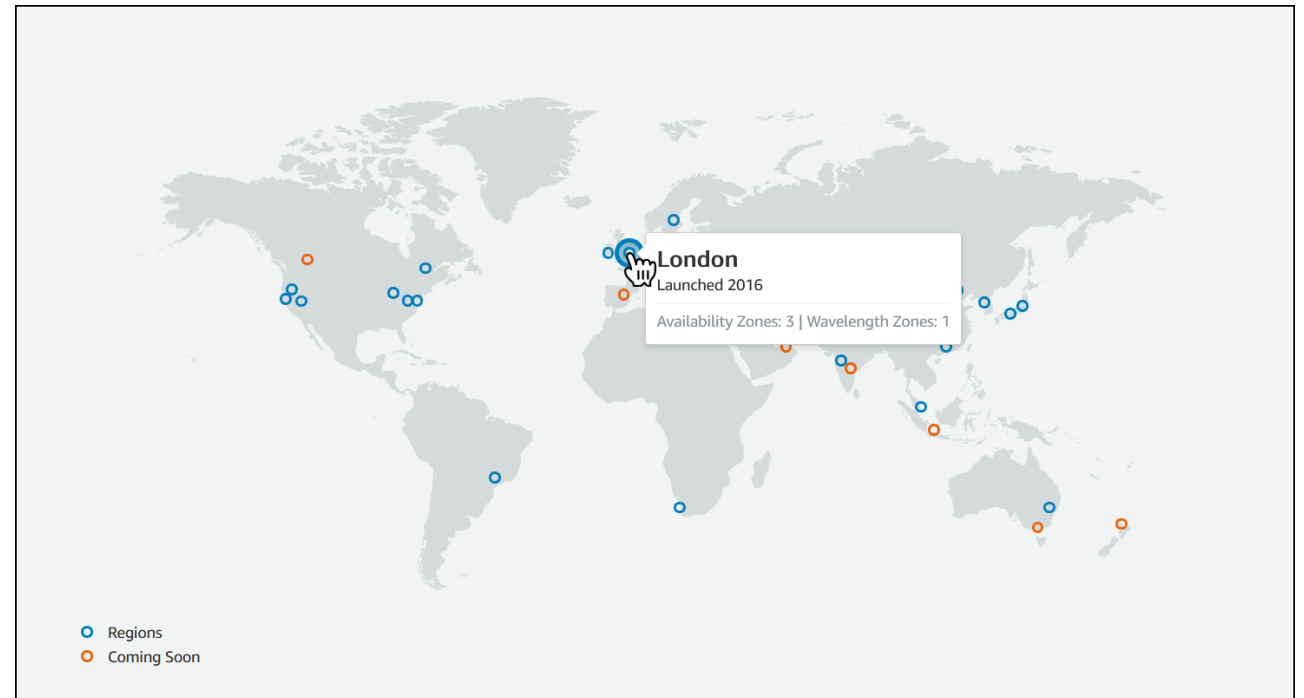


Figure 1.1 AWS data center locations, “Amazon Web Services in Action” by Manning

AWS Global Infrastructure

- The **AWS Global Infrastructure** is designed and built to deliver a **flexible**, **reliable**, **scalable**, and **secure** cloud computing environment with high-quality **global network performance**.
- AWS continually updates its global infrastructure footprint. Visit one of the following web pages for current infrastructure information:

- AWS Global Infrastructure Map:
https://aws.amazon.com/about-aws/global-infrastructure/#AWS_Global_Infrastructure_Map
Choose a circle on the map to view summary information about the Region represented by the circle.
- Regions and Availability Zones:
https://aws.amazon.com/about-aws/global-infrastructure/regions_az/
Choose a tab to view a map of the selected geography and a list of Regions, Edge locations, Local zones, and Regional Caches.



AWS Regions

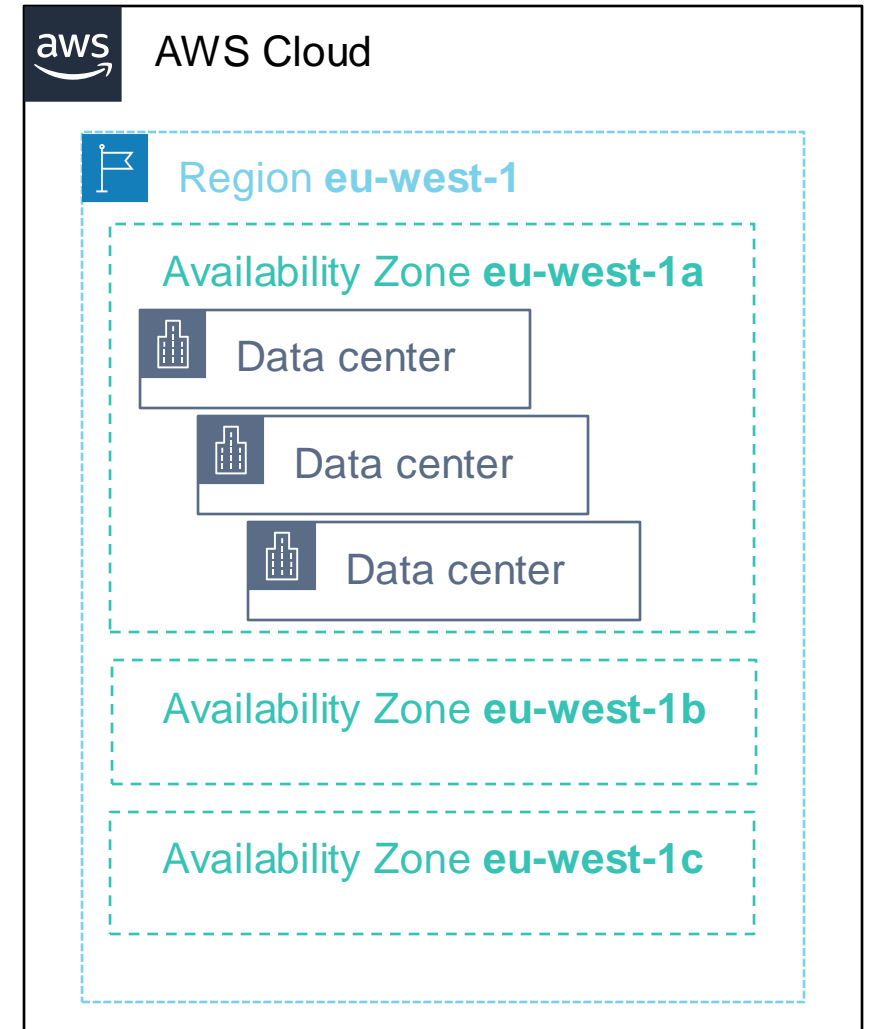
- An **AWS Region** is a geographical area.
 - **Data replication** across Regions is controlled by you.
 - **Communication** between Regions uses AWS backbone network infrastructure.
- Each Region provides full redundancy and connectivity to the network.
- A Region typically consists of two or more **Availability Zones**.



Example: London Region

Availability Zones

- Each **Region** has multiple Availability Zones.
- Each **Availability Zone** is a fully isolated partition of the AWS infrastructure.
 - Availability Zones consist of discrete **data centers**
 - They are designed for fault isolation
 - They are interconnected with other Availability Zones by using high-speed private networking
 - You choose your Availability Zones.
 - **AWS recommends replicating data and resources across Availability Zones** for resiliency.



AWS Sample Applications

E-commerce Site

Bare minimum setup

- Requirements
 - A web server handles requests from customers
 - A database stores product information and orders
- “On-premise” setup
 - Rent servers in a data center
- Cloud/AWS setup
 - Using virtual machine (e.g. AWS EC2) as web server
 - Cloud hosted database (e.g. AWS RDS) as database

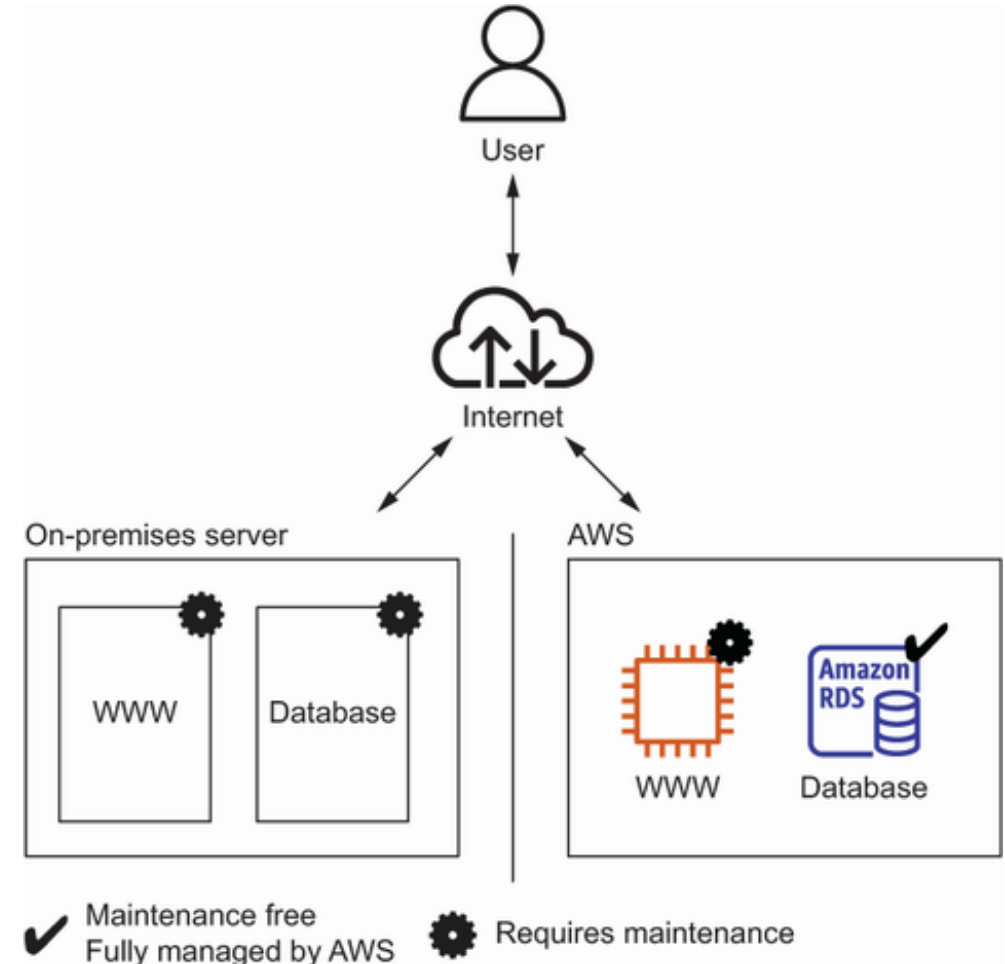


Figure 1.2 Running a web shop on-premises vs. on AWS

Enhanced 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).
- Run the web server on multiple virtual machines to achieve high availability and to reduce response time

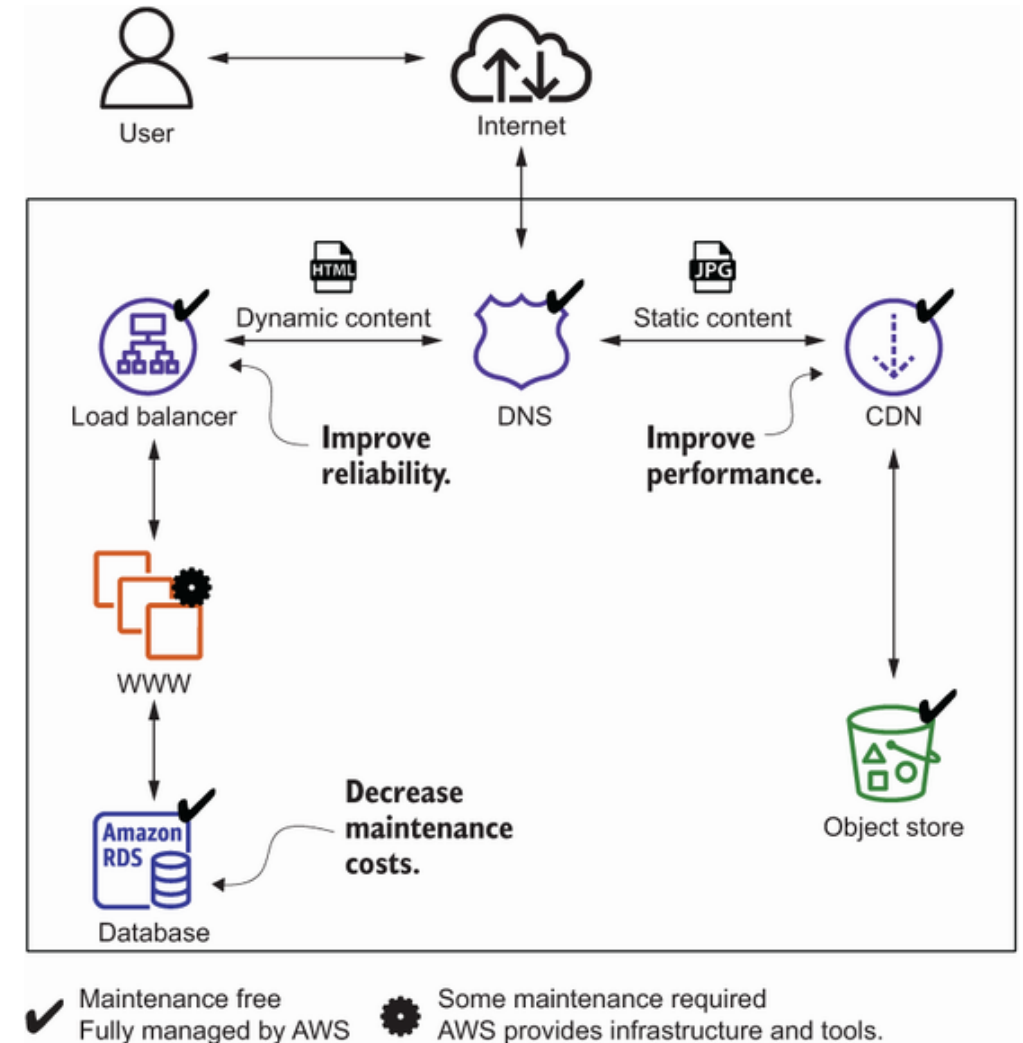


Figure 1.3 Running a web shop on AWS with CDN for better performance, a load balancer for high availability, and a managed database to decrease maintenance costs

HA to the next level

- DB replica across data centers
- Multiple virtual machines as web server across data centers
- Load balancer to distribute customer requests

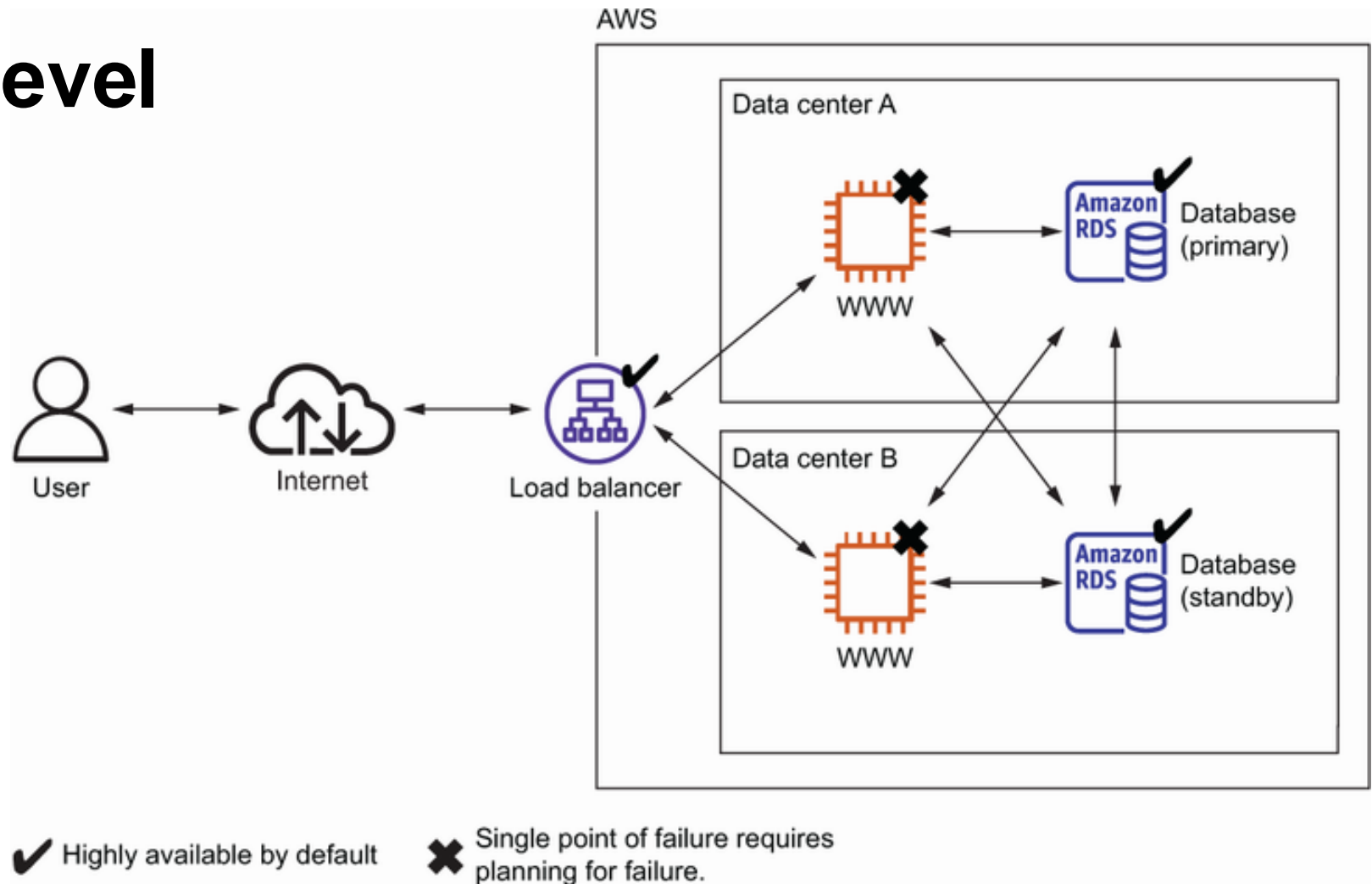


Figure 1.5 Building a highly available system on AWS by using a load balancer, multiple virtual machines, and a database with primary-standby replication

Batching Processing Infrastructure

Scenario



Nick is a data scientist who needs *to process massive amounts of measurement data* collected from gas turbines. He needs to generate *a daily report* containing the maintenance condition of hundreds of turbines. 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.

- Chapter 1 “Amazon Web Services in Action” by
Manning

Cloud based cost effective solution

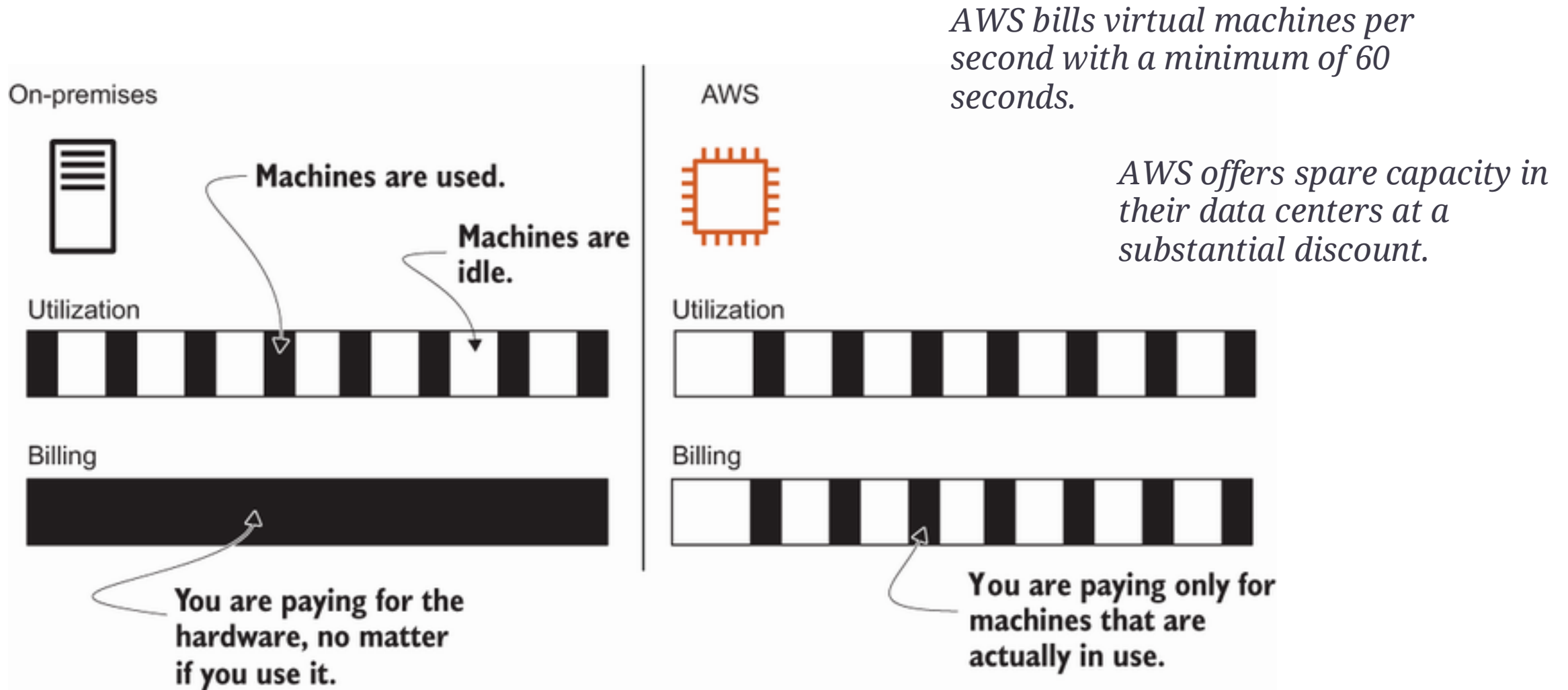


Figure 1.6 Making use of the pay-per-use price model of virtual machines

Cost of AWS

Free tier

- Free tier services for new account within the first 12 months of signing up
 - Limited services types
- Education credit/access
 - We use this option to do lab exercises and assessments

Billing example

- Three major billing categories
 - Based on time of use
 - Based on traffic
 - Based on storage usage
- There are other quality based charging

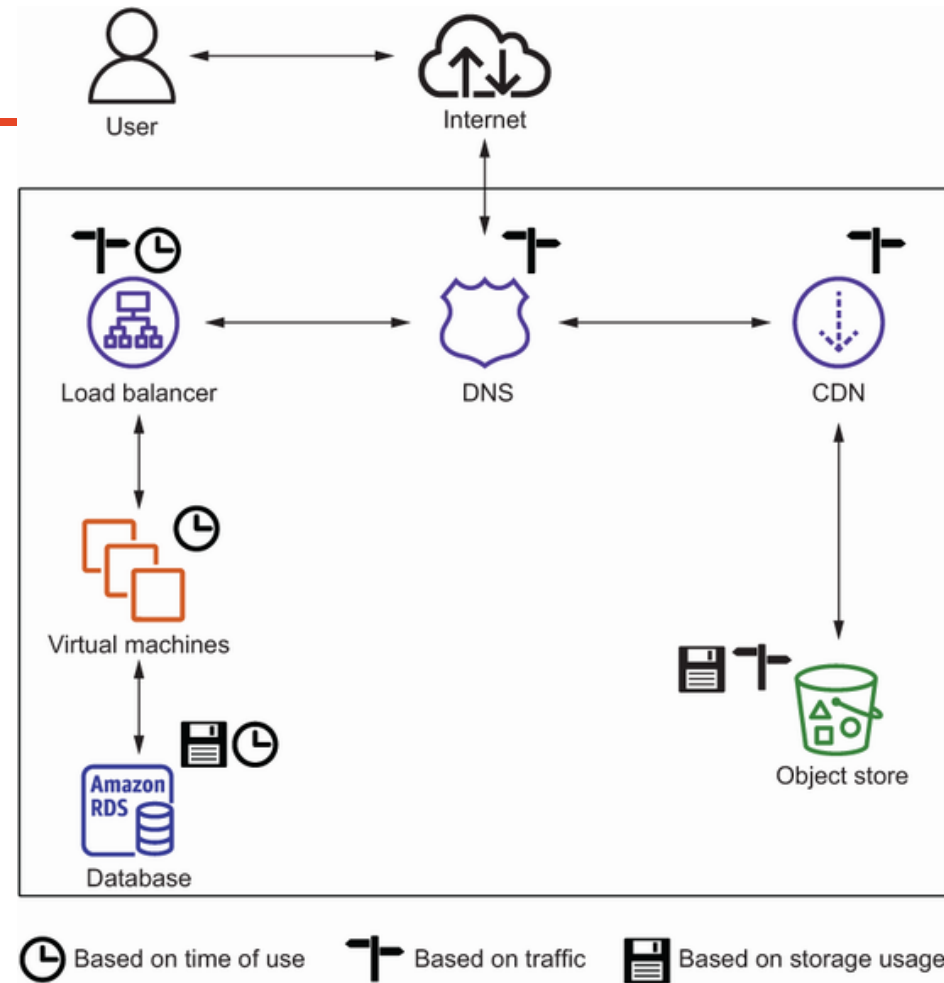


Figure 1.8 Some services are billed based on time of use, others by throughput or consumed storage.

Usage and Costs

- A web shop started successfully in January
- The number of visitors to the web shop increased fivefold in February due to successful campaign
- The cloud cost is likely to increase linearly depending on the actual usage

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	\$115.00	\$100.00
Static files	50 GB used storage	50 GB used storage	\$1.15	\$0.00
Load balancer	748 hours + 50 GB traffic	748 hours + 250 GB traffic	\$19.07	\$1.83
Web servers	1 virtual machine = 748 hours	4 virtual machines = 2,992 hours	\$200.46	\$150.35
Database (748 hours)	Small virtual machine + 20 GB storage	Large virtual machine + 20 GB storage	\$133.20	\$105.47
DNS	2 M requests	10 M requests	\$4.00	\$3.20
Total cost			\$472.88	\$360.85

Exploring AWS Services

The Overall Picture

- Services are created and managed by sending requests to the corresponding API
 - Through a web-based GUI like management console
 - Command line interface like AWS CLI
 - Programmatically via SDK
- Virtual machines can be accessed through SSH and can be managed in the same way as a physical server
- Majority of the services are behind the APIs

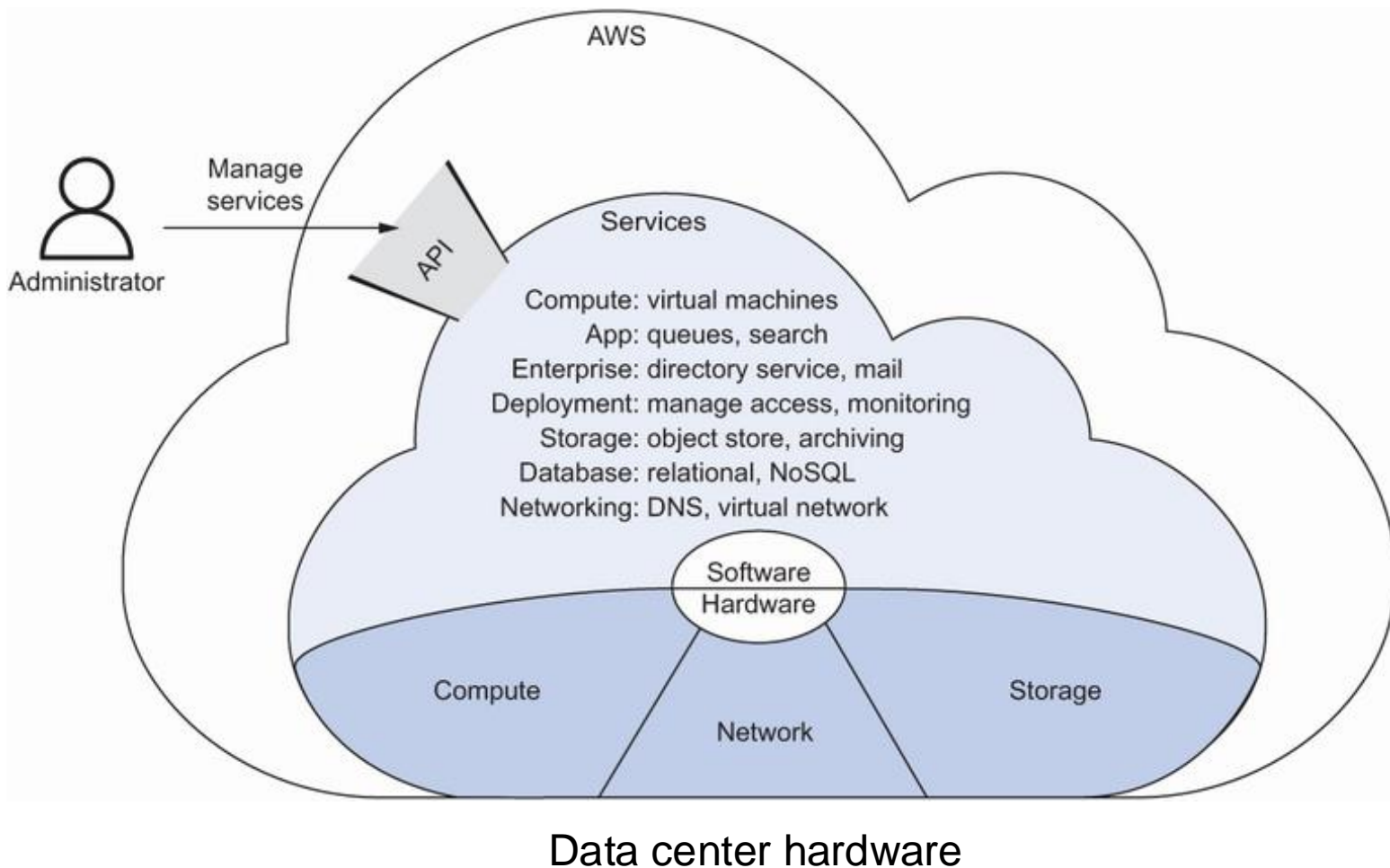
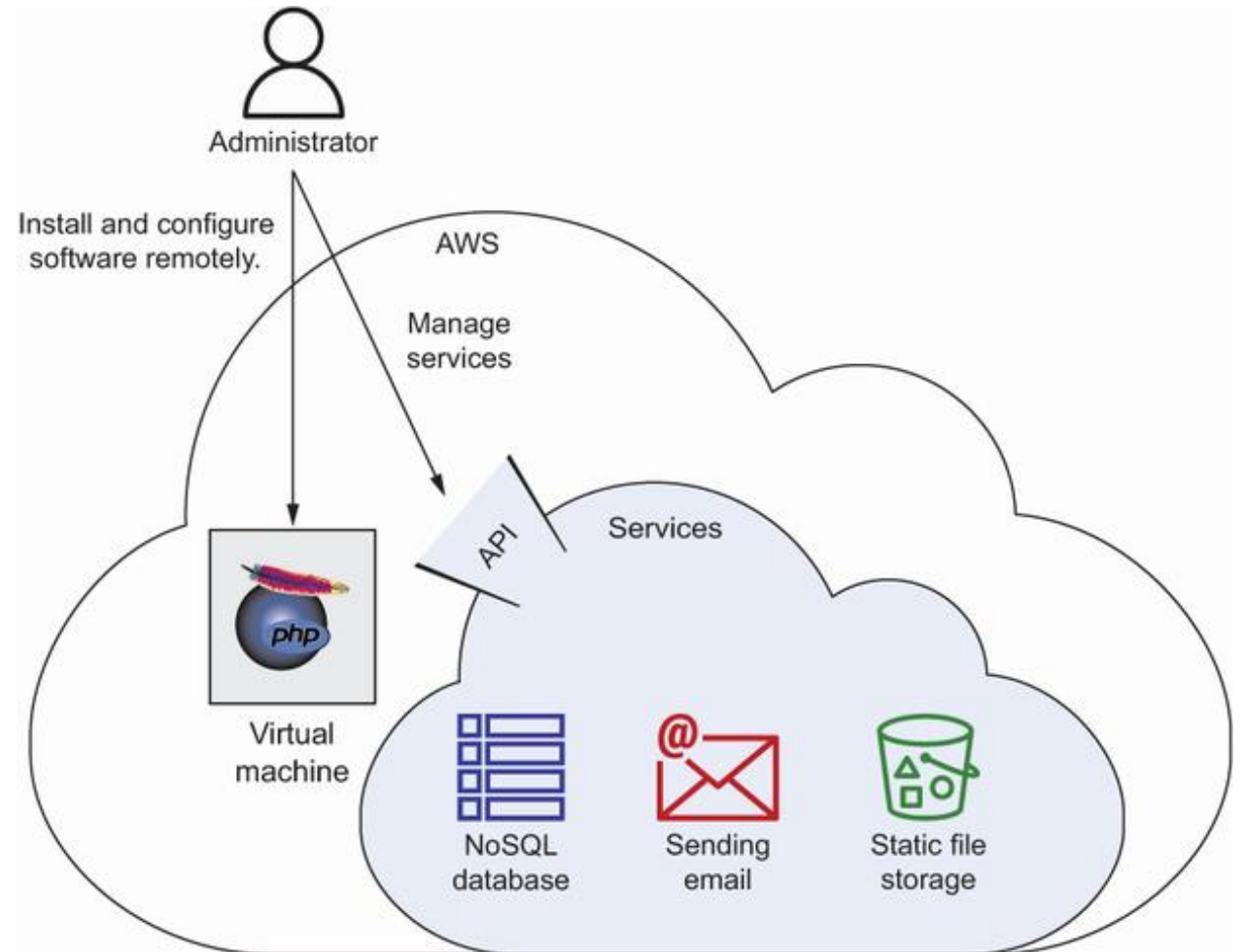


Figure 1.9 The AWS cloud is composed of hardware and software services accessible via an API.

Managing a Simple Web Application

- The administrators use AWS APIs to create/configure necessary services
- The virtual machine can be setup further through SSH
 - Uploading web server code
 - Configuring parameters



From End User Perspective

- The VM is the front end
- They send HTTP requests to the VM, which runs a web server along with a custom PHP web application.
- The web application talks to AWS services to answer HTTP requests from users
 - query data from a NoSQL database,
 - store static files
 - send email.
- Communication between the web application and AWS services is handled by the API,

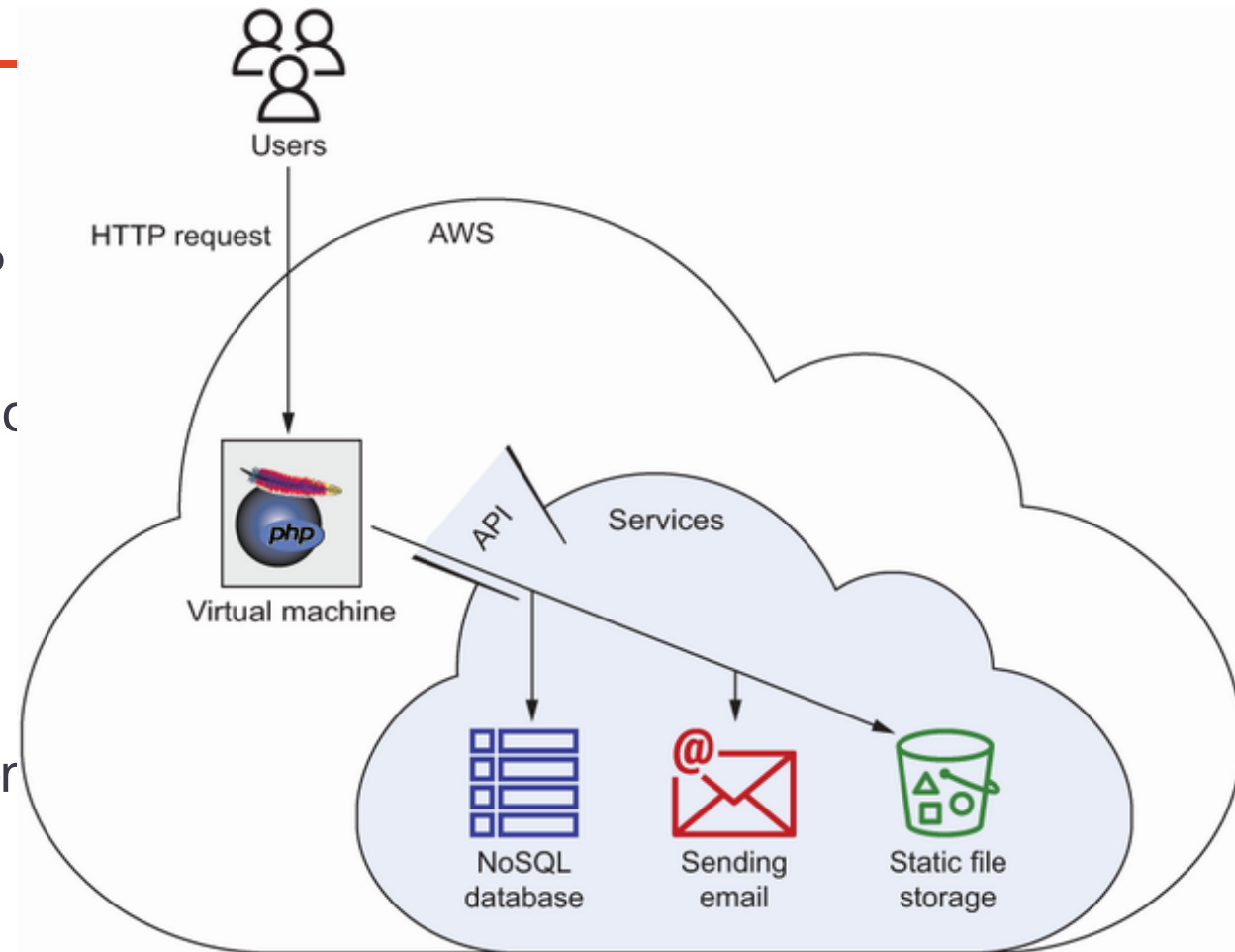


Figure 1.11 Handling an HTTP request with a custom web application using additional AWS services

Interacting with AWS Services

Four options of interacting with AWS

- They represent different end user interfaces of the same API
 - Management Console
 - Command line
 - SDK
 - Blueprints

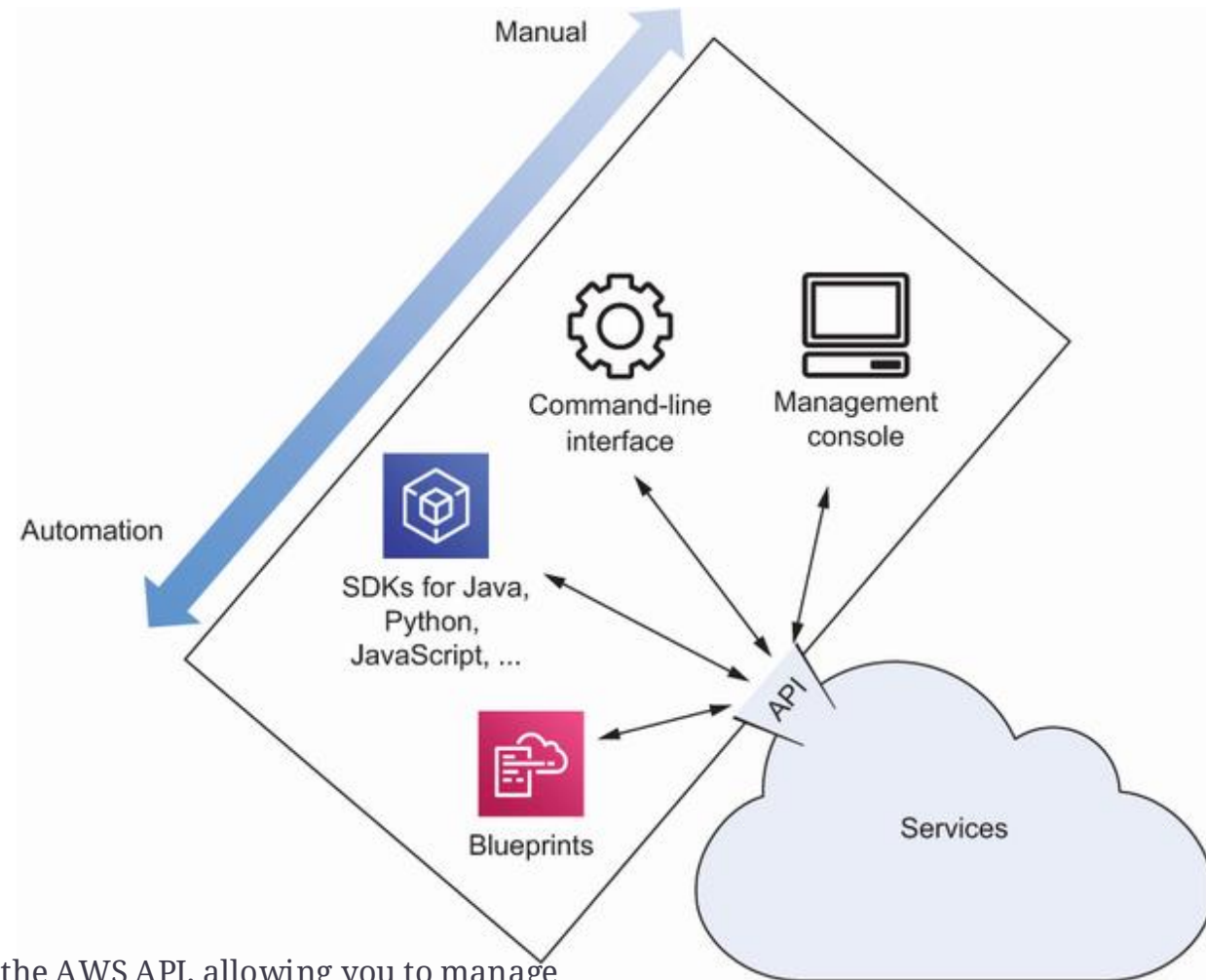


Figure 1.12 Different ways to access the AWS API, allowing you to manage and access AWS services

Management Console

- The starting point for nearly all users
- Easy to use
- Best for setup simple infrastructure for development and testing

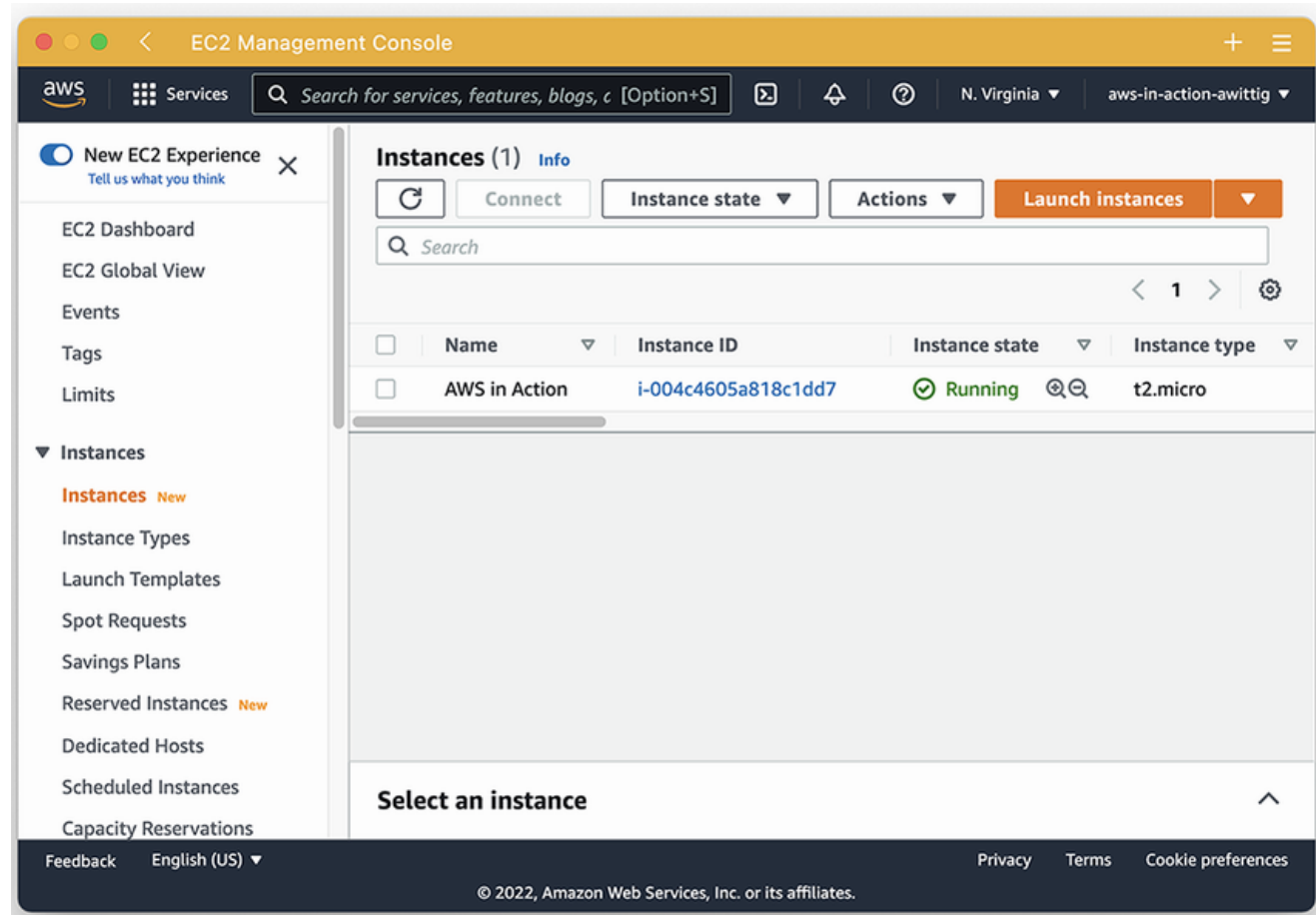
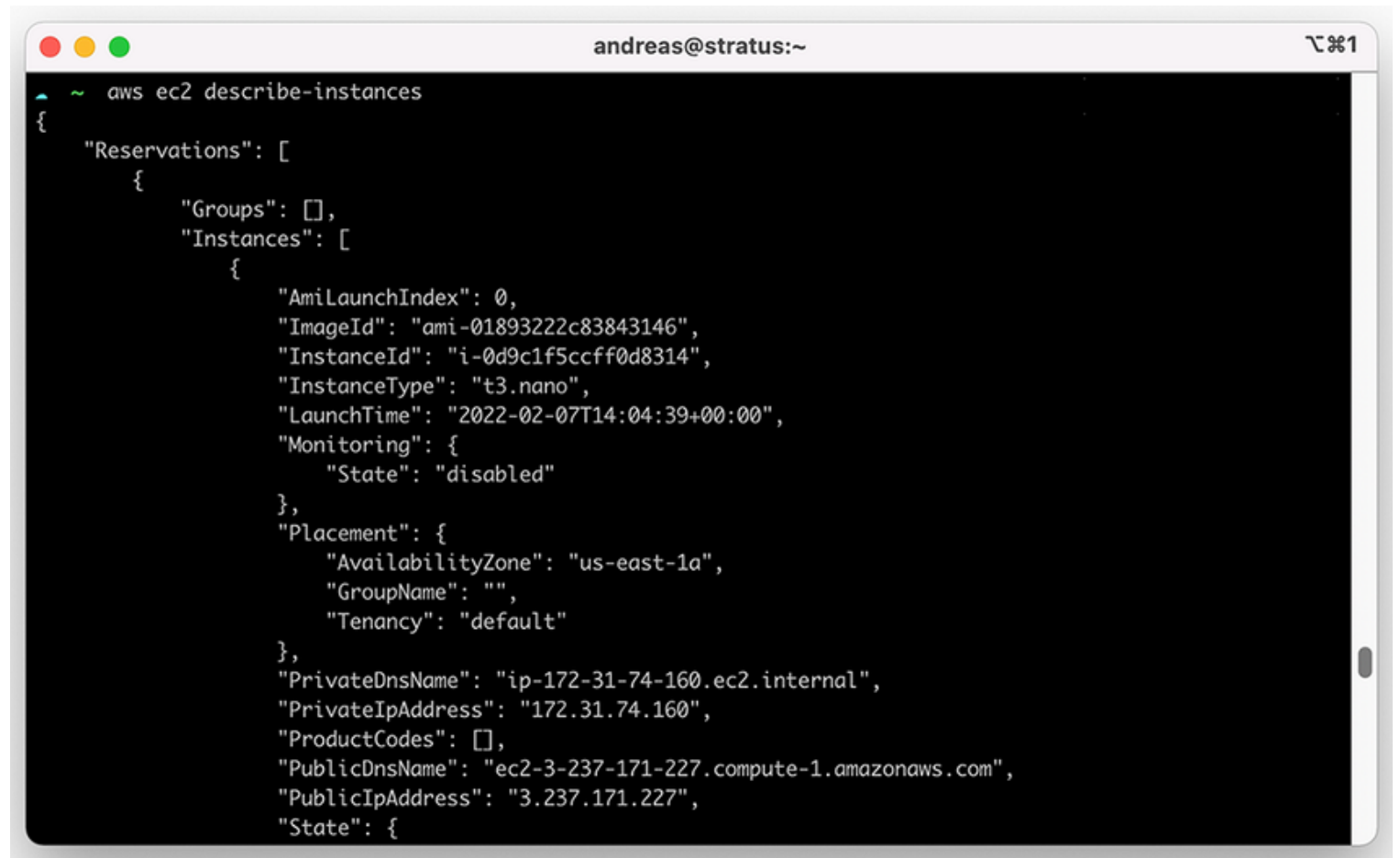


Figure 1.13 The AWS Management Console offers a GUI to manage and access AWS services.

Command-line Interface

- Allows user to manage and access AWS services within their terminal
- Best for automate or semi-automating recurring tasks.
- Typical use cases
 - Create new infrastructure based on blueprint
 - Upload files
 - Inspect services

A terminal window titled 'andreas@stratus:~' with a dark background and light green text. The command 'aws ec2 describe-instances' has been executed, resulting in a JSON output. The output shows a list of reservations, with the first reservation containing details for a single instance: 't3.nano' type, launched on '2022-02-07T14:04:39+00:00', in the 'us-east-1a' availability zone, with a public IP address of '3.237.171.227'.

```
andreas@stratus:~  
~ aws ec2 describe-instances  
{  
  "Reservations": [  
    {  
      "Groups": [],  
      "Instances": [  
        {  
          "AmiLaunchIndex": 0,  
          "ImageId": "ami-01893222c83843146",  
          "InstanceId": "i-0d9c1f5ccff0d8314",  
          "InstanceType": "t3.nano",  
          "LaunchTime": "2022-02-07T14:04:39+00:00",  
          "Monitoring": {  
            "State": "disabled"  
          },  
          "Placement": {  
            "AvailabilityZone": "us-east-1a",  
            "GroupName": "",  
            "Tenancy": "default"  
          },  
          "PrivateDnsName": "ip-172-31-74-160.ec2.internal",  
          "PrivateIpAddress": "172.31.74.160",  
          "ProductCodes": [],  
          "PublicDnsName": "ec2-3-237-171-227.compute-1.amazonaws.com",  
          "PublicIpAddress": "3.237.171.227",  
          "State": {
```

SDKs

- Language specific SDKs wrap up the AWS APIs so that AWS services can be integrated in applications conveniently
 - E.g. integrating AWS database service in a desktop application
- These are the supported languages

• JavaScript • Python • PHP

• .NET • Ruby • Java

• Go • Node.js • C++

Blueprints

- A *blueprint* is a description of a 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.
 - Amazon CloudFormation
 - Terraform

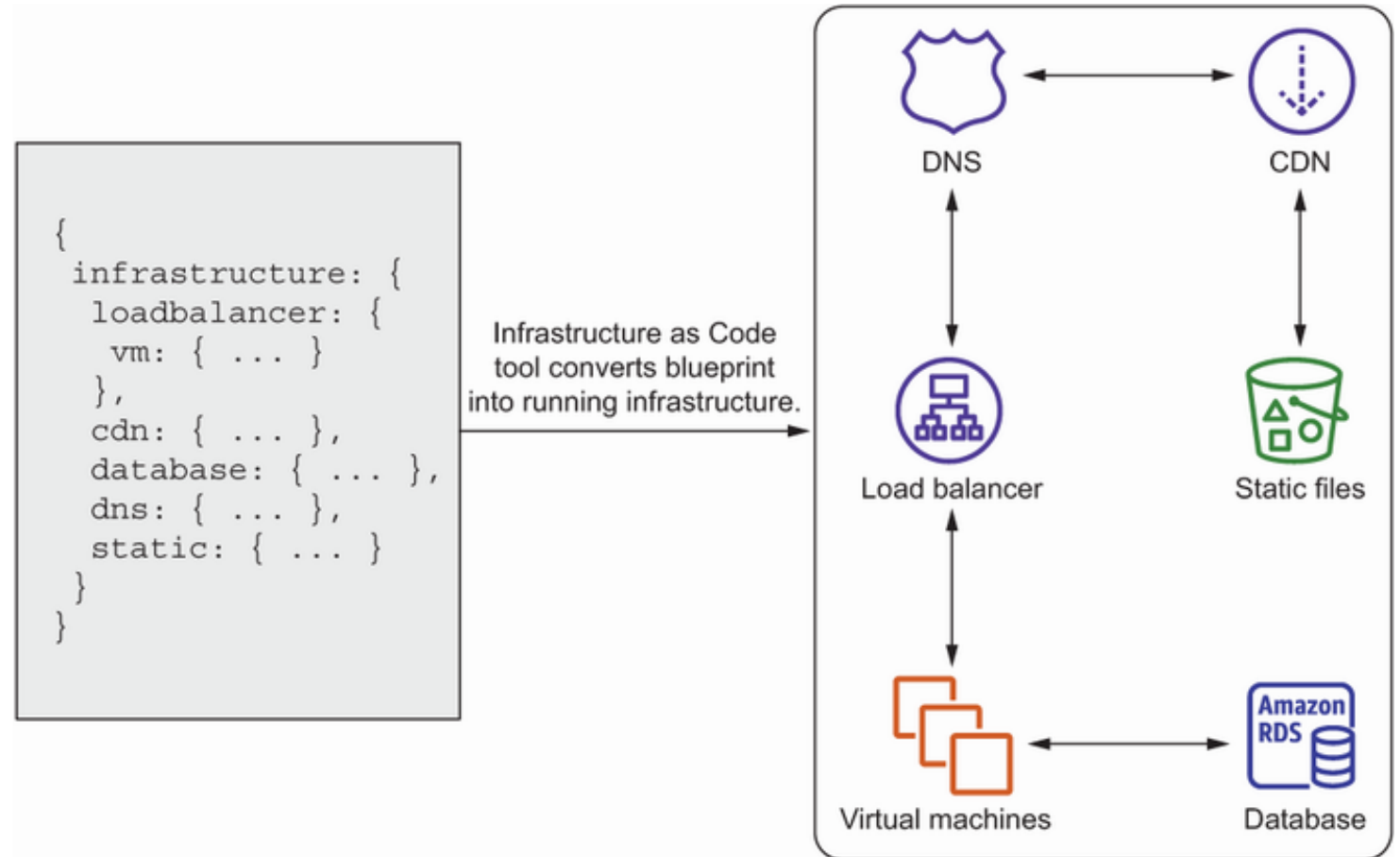


Figure 1.15 Infrastructure automation with blueprints

Accessing AWS Services

AWS Account and IAM services

- In general, you will need an AWS account to be able to start using any AWS services, including the free-tier ones
- But, there are other ways to gain access to AWS resources
 - E.g. all employees of a company could use the same account to access AWS resources
 - They don't use the actual account name and password to login
 - What could be the issue of that?
 - Most cloud platforms provide an Identity and Access Management (IAM) service to handle such and other access scenarios
- In this unit, we obtain access of limited AWS services free of charge through third party authentication
 - We login to AWS Academy Canvas
 - Within a lab environment, we gain access to *limited* AWS resources
 - Each lab environment gives us access to different types of resources
 - An error message will display when trying to access resources
 - Sometimes, an error message may be displayed automatically on part of the UI, we can ignore that

IAM: Essential components



IAM user

A **person** or **application** that can authenticate with an AWS account.



IAM group

A **collection of IAM users** that are granted identical authorization.



IAM policy

The document that defines **which resources can be accessed** and the **level of access** to each resource.



IAM role

Useful mechanism to grant a set of permissions for making AWS service requests.

Authenticate as an IAM user to gain access

When you define an **IAM user**, you select what *types of access* the user is permitted to use.

Programmatic access

- Authenticate using:
 - Access key ID
 - Secret access key
- Provides AWS CLI and AWS SDK access



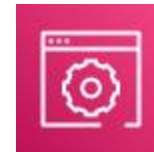
AWS CLI



AWS Tools
and SDKs

AWS Management Console access

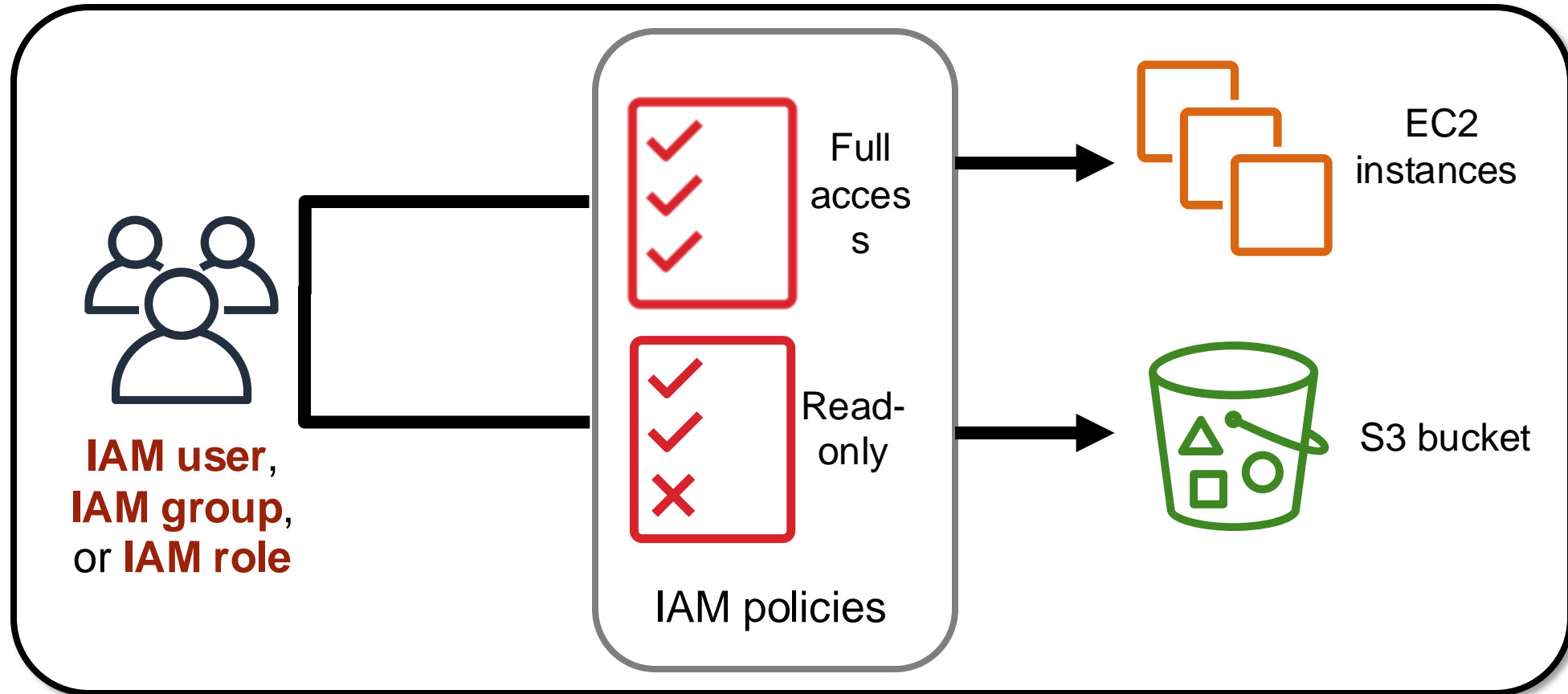
- Authenticate using:
 - 12-digit Account ID *or* alias
 - IAM user name
 - IAM password
- If enabled, **multi-factor authentication (MFA)** prompts for an authentication code.



AWS Management
Console

Authorization: What actions are permitted

After the user or application is connected to the AWS account, what are they allowed to do?



IAM: Authorization

- Assign permissions by creating an IAM policy.
- Permissions determine **which resources and operations** are allowed:
 - All permissions are implicitly denied by default.
 - If something is explicitly denied, it is never allowed.

Best practice: Follow the **principle of least privilege**.

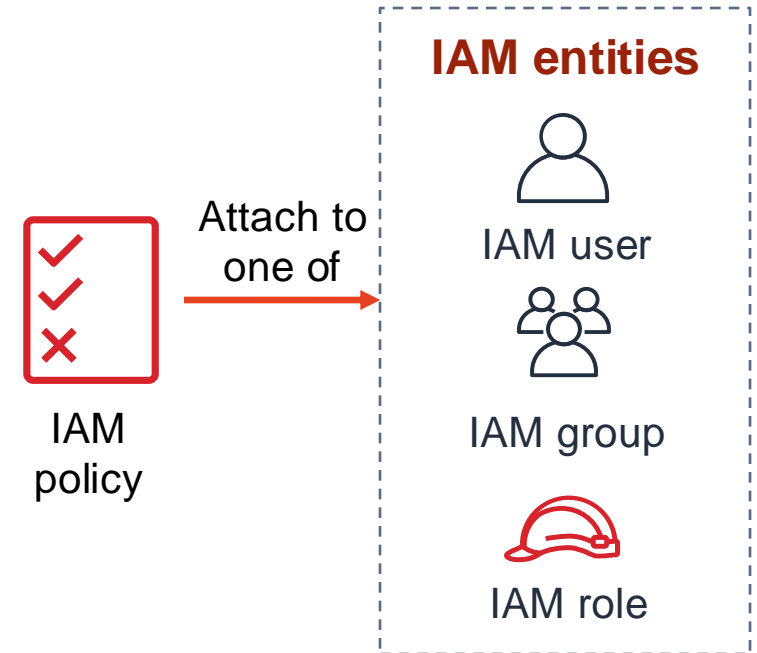


**IAM
permissions**

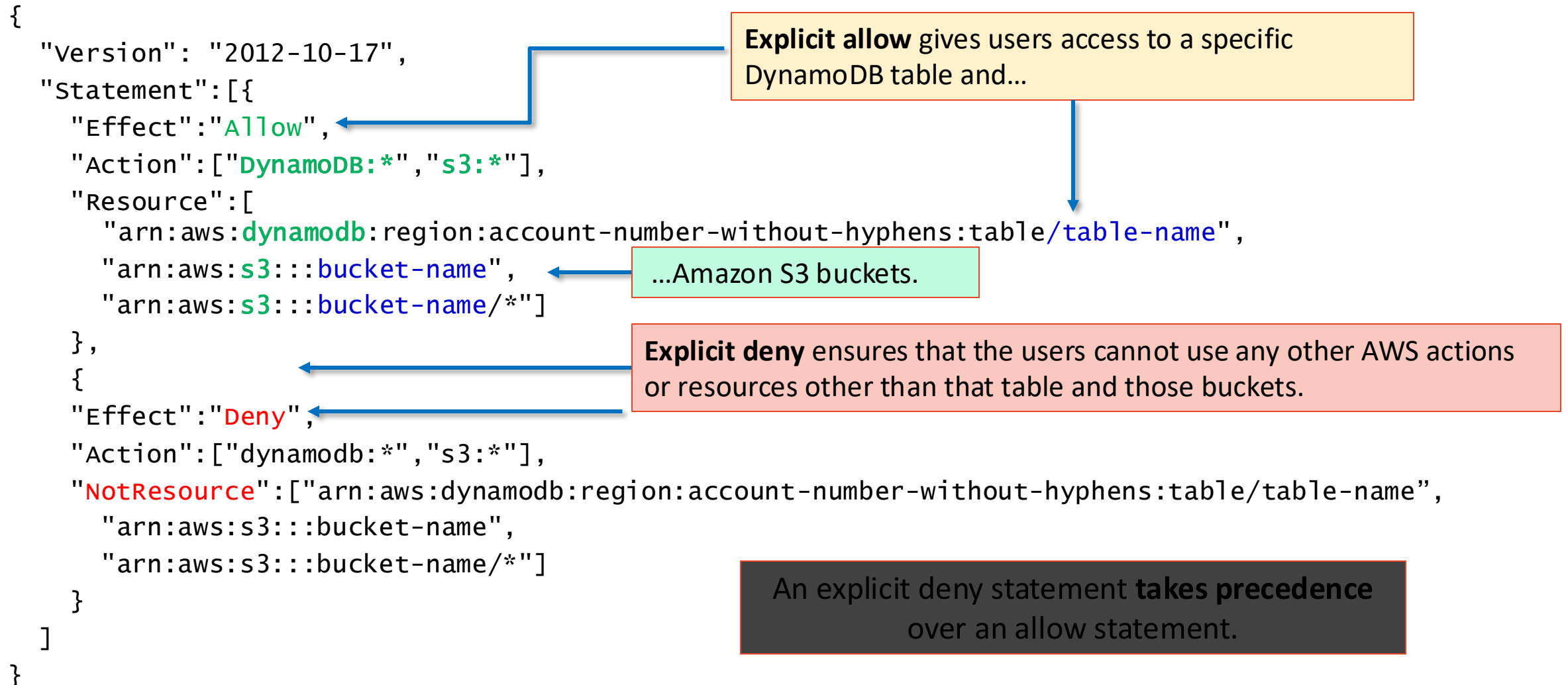
Note: The scope of IAM service configurations is **global**. Settings apply across all AWS Regions.

IAM policies

- **An IAM policy is a document that defines permissions**
 - Enables fine-grained access control
- Two types of policies – *identity-based* and *resource-based*
- **Identity-based** policies –
 - Attach a policy to any IAM entity
 - An IAM user, an IAM group, or an IAM role
 - Policies specify:
 - Actions that **may** be performed by the entity
 - Actions that **may not** be performed by the entity
 - A single *policy* can be attached to multiple *entities*
 - A single *entity* can have multiple *policies* attached to it
- **Resource-based** policies
 - Attached to a resource (such as an S3 bucket)

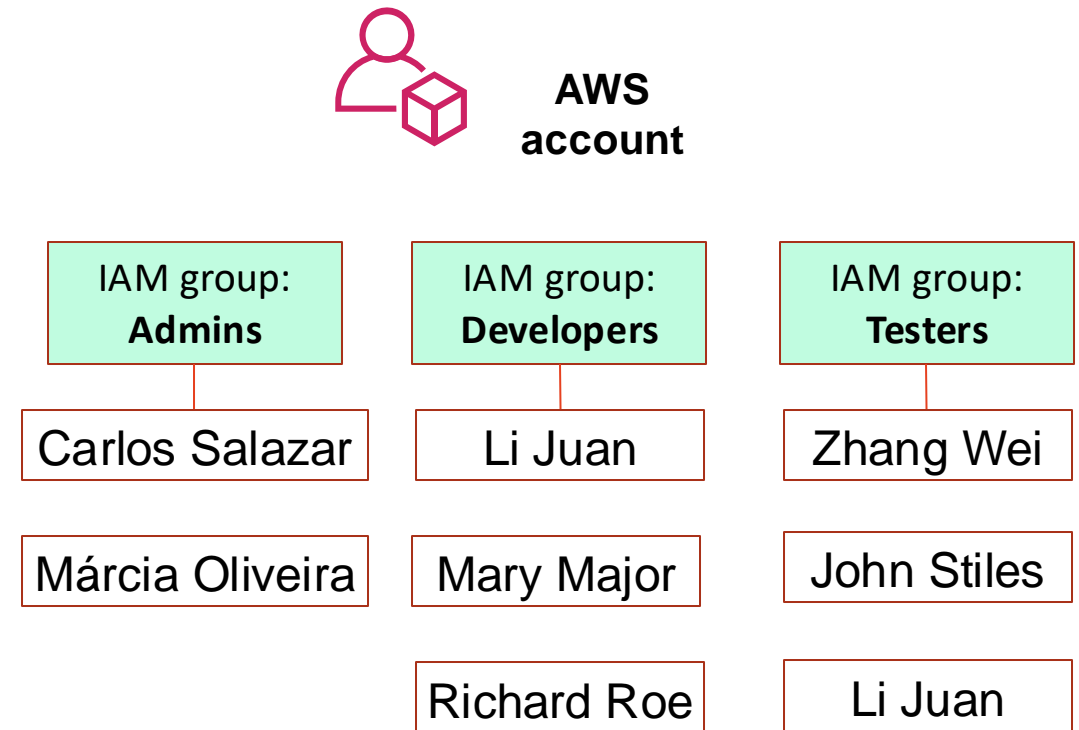


IAM policy example



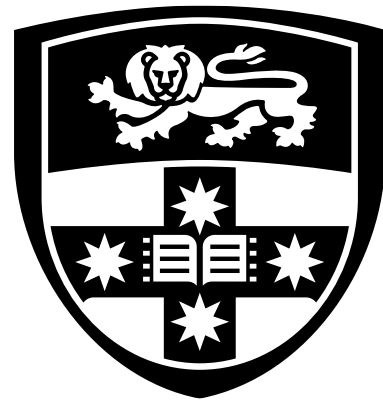
IAM groups

- An **IAM group** is a collection of IAM users
- A group is used to grant the same permissions to multiple users
 - Permissions granted by attaching IAM *policy* or policies to the group
- A user can belong to multiple groups
- There is no default group
- Groups cannot be nested



References

- Michael Wittig, Andreas Wittig, **Amazon Web Services in Action, Third Edition**
 - Chapter 1



THE UNIVERSITY OF
SYDNEY