

Networks and Internet Applications

Continuous Assessment Test – CA2

Nicolas D'Alessandro Calderon

Answers

1. You have been asked to strategically evaluate the adoption of cloud computing at a telecommunications company that is considering migrating part of its infrastructure to the cloud. They currently have their own infrastructure with physical servers that host their communications services and customer data storage.
 - a. What are the main advantages and disadvantages of migrating to the cloud for this company in terms of security, costs and performance? Include at least two advantages and two disadvantages for each aspect.
 - b. Propose a deployment model (public, private, hybrid, or community) that best suits this company and justify your choice.
 - c. Considering the possibility of adopting a hybrid strategy, which parts of their infrastructure should they keep on-premises and which ones should they outsource, taking into account the following aspects: latency and performance; regulations and privacy; availability?

To answer this question, I will first elaborate a detailed analysis applying some of the tools and strategies used in real scenarios to evaluate and define the cloud adoption and migration analysis:

Cloud Migration Analys

Security comparative table

Criteria	On-premises	Public Cloud	Private Cloud	Hybrid Cloud
Data control	★★★★ High	★★ Low	★★★★ High	★★★ Moderate
Cyberattacks protection	Depending on the internal resources	★★★★ Advanced	★★★★ Advanced	★★★★ Advanced
Regulatory compliance	Direct control	Harder - Complex	Often Easier - Simple	Adaptable
Response to incidents	Limited to the own resources	Professional	Mixed	Complete

Costs comparative table

Criteria	On-premises	Public Cloud	Private Cloud	Hybrid Cloud
Initial Investment	★★★★ High - CapEx	★★ Low	★★★★ Moderate to High	★★★ Moderate
Operative Costs	Fixed + Variables	Variable - Opex	Middle	Mixed
Scalability	Expensive	Cheap	Middle - Adaptable	Efficient
ROI	Long-term	Short-term	Mid-term	Can be optimized

Performance comparative table

Criteria	On-premises	Public Cloud	Private Cloud	Hybrid Cloud
Latency	★★★★ Low	★★★ Variable	★★★★ Low	★★★ Can be optimized
Scalability	★ Limited	★★★★★ High	★★★★ High	★★★★ High
Resource Control	★★★★★ Total	★★ Limited	★★★★ High	★★★ Selective
Availability	Limited to the own resources	★★★★ High	★★★★ High	★★★★★ Very High

- a. Based on this analysis we can identify the following advantages and disadvantages in each category when migrating to the cloud:

Security

Advantages:

- Access to advanced technologies and specialized equipment to ensure optimal security in our network.
- Advanced protection to Cyberattacks, given that many of the cloud providers offer constant monitoring combined with advanced systems already mentioned.

Disadvantages:

- Limited control and visibility over the data and the physical infrastructure where the data lives.
- Complexity on data regulation control such as GDPR and data governance.

Costs

Advantages:

- The Capex model is replaced by an Opex model, allowing the company to reduce initial investment costs by avoiding the purchase and maintenance of physical servers.
- Pay per use model allows the company to scale the resources dynamically and based on the demand, significantly improving its financial performance.

Disadvantages:

- Hidden costs for scalability that sometimes are not easy to control may result in high bills.
- For a telecommunication company, given the need for high-volume data transfer, cost may be significant if it is not correctly managed.

Performance

Advantages:

- High availability and scalability with architectures that efficiently manage the load balancing, and the global availability, improving uptime, response capacity, and allowing the company to adapt to unexpected demand peaks.
- Instant access to a global infrastructure helps to improve the user experience.

Disadvantages:

- Communication services in the cloud may be affected by additional latency if not managed correctly.
- There may be a limitation on customization options and a high dependency on the internet connection that in case of any problem, will impact directly in the service.

b. A **Hybrid Cloud Deployment** model seems to be the most suitable for our use case:

- This model combines the **control and security of on-premises** infrastructure with the **flexibility and scalability of cloud services**.
- It will also allow the company to **keep sensible customers that need special or real-time services**, ensuring compliance with data protection and regulations and maintaining low latency.
- Critical services related to data analytics, backups, or CRM systems can be **easily migrated to the public cloud to benefit from the cost saving and scalability**.
- It will allow a **gradual and controlled migration** minimizing operational risks.

c. Suggested Infrastructure allocation:

Aspect	Keep on-premises	Migrating to the cloud
Latency and performance	Core communication services such as voice/data/video or routing systems that require real-time low latency.	Analytics, databases and big data processing that do not require real time data. Web portals for the clients and mobile applications. Back-office systems.
Regulations and privacy	Sensitive customer data, confidential business information, systems subject to strict regulatory compliance (GDPR, financial, health data, etc.).	Anonymized data for marketing analysis Development and testing environments. Public content and marketing material.
Availability	- Critical infrastructure that is necessary during connectivity issues. Primary services for authentication and access control. Critical systems with service level agreements guarantees.	Services for redundancy, backups, disaster recovery, etc. Elastic resources to manage demand peaks

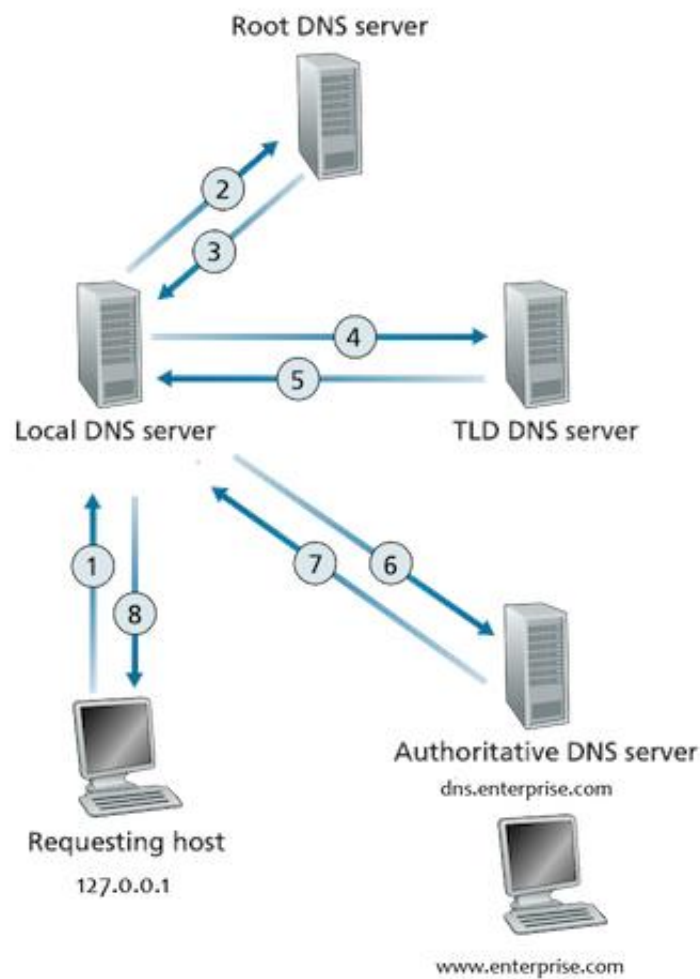
2. Suppose you want to connect to **www.enterprise.com** and the following records are on the DNS server for the TLD (Top Level Domain):

- (www.enterprise.com, dns.enterprise.com, NS)
- (dns.enterprise.com, 146.54.104.222, A)

Also suppose the following records are on the DNS server for enterprise.com:

- (www.enterprise.com, east5.enterprise.com, CNAME)
- (east5.enterprise.com, 142.81.17.206, A)
- (enterprise.com, mail.enterprise.com, MX)
- (mail.enterprise.com, 247.29.174.157, A)

Also, suppose your local DNS server only caches the DNS server for the TLD.



a. In the example above, how many different types of Resource Records (RRs) are there on the authoritative DNS server for enterprise.com?

- **CNAME or Canonical Name Record** that is used for aliasing names. In our example [www.enterprise.com](#), [east5.enterprise.com](#), **CNAME**.
- **A or Address Record** that is used for mapping the hostname to an IPv4 address. In our example [east5.enterprise.com](#), [142.81.17.206](#), **A**, and [mail.enterprise.com](#), [247.29.174.157](#), **A**.
- **MX or Mail Exchange Record** that is used to specifies the mail server responsible for the domain [enterprise.com](#), [mail.enterprise.com](#), **MX**.

b. Can you send multiple DNS queries and receive multiple RR responses in a single message?

Yes, it is possible since DNS protocol supports sending multiple queries (using EDNS(0) or extension) and also a single DNS query may return multiple Resource Records.

So, as an example a client may send a unique DNS query asking information to [www.enterprise.com](#) and the server will retrieve in only one answer different related Resource Records: A with the IP address, CNAME that points to other name and the TTL of these records.

c. To which DNS server does the host send its requests?

The host is not contacting directly to the internet but first asking to the local DNS server or resolver (corresponding to the arrow 1 of our diagram). This local DNS handles the recursive resolution process asking to the root servers (2,3), then TLD (4,5), Authoritative (6,7) and so on.

d. What type of DNS server stores a company's DNS records?

The company records are stored in the authoritative DNS server, which is the company server and the “authorized” to answer about this domain. In this example [dns.enterprise.com](#) is the authoritative DNS for the [enterprise.com](#) domain.

e. What is the DNS server name for enterprise.com?

As we explain before the DNS Server Name is [dns.enterprise.com](#).

f. When you connect to [www.enterprise.com](#), your local DNS server requests the IP address on your behalf. When it contacts the TLD server, how many RRs are returned? Indicate the content of these responses.

So, when the DNS local server asks the TLD sever, this returns two RRs:

1. [www.enterprise.com](#), [dns.enterprise.com](#), **NS** meaning to “get to know” about enterprise.com “talk to” [dns.enterprise.com](#)
2. [dns.enterprise.com](#), [146.54.104.222](#), **A** meaning “the IP of [dns.enterprise.com](#) is 146.54.104.22.”

g. Suppose the enterprise.com website is hosted at east5.enterprise.com. What type of record is needed to do this?

- A CNAME or Canonical Record is needed for this operation.
- This would be (www.enterprise.com, east5.enterprise.com, CNAME) already present in the authoritative DNS server.
- This record will connect www.enterprise.com with east5.enterprise.com.

h. Suppose we are trying to send an email to admin@enterprise.com. What type of record will contain the name of your mail server and what is its value?

- In this case the MX or Mail Exchange record will contain the name of the mail server.
- The value is mail.enterprise.com present in the record (enterprise.com, mail.enterprise.com, MX).
- The MX record says to our computer “the emails from enterprise.com goes to mail.enterprise.com.

i. Why does the local DNS server cache the TLD's DNS server?

- **To optimize the resolution process and the efficiency in search times.**
- By keeping this information in cache, in future requests, the DNS server can ask directly to the TLD **server without having to query the root DNS servers** every time it needs to resolve a domain name.
- This caching process **reduces the latency and the load in the root servers** improving the overall performance.

j. What transport protocol(s) does DNS use: TCP, UDP, or both?

- DNS protocol **uses both TCP and UDP** depending on the use case or situation:
 - **UDP** is the default for most of the DNS queries and answers because is the faster and efficient for small data.
 - **TCP** is used when the response exceeds the 512 bytes such as zone transfers or when the reliability is more important than the speed of transfer.

k. What well-known port does DNS use?

- DNS uses the **port 53** (for both TCP and UDP). This is the standard port for DNS described by the Internet Assigned Numbers Authority.

- Choose an email you recently received. Make sure it's from the year 2025. Access the option to view the full email header. In most email clients, you can do this by looking for the "View Original" or "Show Header" option.

Original Message

Message ID	<010201966d20033b-afe7ec46-7db2-4b82-b241-4bca35c77f1b-000000@eu-west-1.amazonaws.com>
Created at:	Fri, Apr 25, 2025 at 3:25 PM (Delivered after 1 second)
From:	Software Design Patterns - Classroom 1 <notifications@instructure.com>
To:	nicodalessandro11@gmail.com
Subject:	Solution PR1: Software Design Patterns - Classroom 1
SPF:	PASS with IP 54.240.89.127 Learn more
DKIM:	'PASS' with domain instructure.com Learn more
DMARC:	'PASS' Learn more

```

Delivered-To: nicodalessandro11@gmail.com
Received: by 2002:a85:648c:e:b0:220:6ff4:9db6 with SMTP id en14csp423351ze1b;
    Fri, 25 Apr 2025 06:25:37 -0700 (PDT)
X-Goog-Source: AGHT4IGHv0XIPxHakBj20QUMIEVmproyV/C8eDVaibcylqY410vJKY2XD5gEzPy8Y3t+qLn
X-Received: by 2002:a85:608c:5251:b0:440:6760:d06a with SMTP id 5b1f17b1804b1-440a64c14damr15817155e9.5.1745587537501;
    Fri, 25 Apr 2025 06:25:37 -0700 (PDT)
ARC-Seal: i=1; a=rsa-sha256; t=1745587537; cv=none;
    d=google.com; s=arc-20240605;
    b=10V3dy2FHa3Drr3ym7pF2m/DreGteV7PqVQWqLqdBK3Kq3eQonbfz3/7Hv8F
    0L83R9AL/FdJpKtWgJ25TVnAcly3MdeqpuCk50nyJA16icf5H4K0xQd4gP
    /35h328K5Y4e0e/F4X20hPQW0R53FLPyfNjG2dKJ4N0Q0uQz;fwlR1me1C1700Kp101
    6E4jYkTypeAkR1Yx17z9ndgeH1ASgtfrusAcve41F4dFmBhkkatexSOVceGmXAG6S
    ES8U5yJzEknghn3yYKcOvTu/HPZ01AShUz6JAE8KshuRkv9zudZYnLQ6eU511REZn
    E984=
ARC-Message-Signature: i=1; a=rsa-sha256; c=relaxed/relaxed; d=google.com; s=arc-20240605;
    h=feedback-id:auto-submitted:content-transfer-encoding:mime-version
    :subject:message-id:to:reply-to:from:date:dkim-signature
    :dkim-signature;
    bh=u3yKsAdp199vU9Rsu1ICoQ0L4Ukufcqh7P48r1Zs=;
    fh=hwjBjBpCYCQV450zIA2PAAY7151U/mfAB5241C7qY=;
    b=BJR4uA8QCP54K5u0dyXp2nQBUU/0pk1ZxifrcdKd357g2EVQqk+8Bv18F
    3NafSGaIR54VzQz05Ad161BLvFY3X8zq2BakPSLo5uawZyA1mnmDFLSYC2VLq/
    T4/Ou1G+0uQJLkTashtH0AQLn0nbn1k848gFV7Tfsc0Wzpho1fJX130w7y0HE
    ME1pnuP77P6t0XqPz12Jh0v0n5f8R2Z45J2jfh117+1127Cvdydpm0aRz
    rnhhg4X0e16EhJG1C3bz530uP011/3h2q4V7Nqhu2A1gs5Nzceq5H0fU0N0E1aCbJ
    k1AA=
ARC-Authentication-Results: i=1; mx.google.com;
    dkim=pass header.i=@instructure.com header.s=hrt54zqnosd2q2fn3sy2vc6aumude2q header.b=HfF9WJ3M;
    dkim=pass header.i=@amazon.com header.s=lnchhvbugj3syuhssfvqohv723u4th header.b=UFA1uK6U;
    spf=pass (google.com: domain of 010201966d20033b-afe7ec46-7db2-4b82-b241-4bca35c77f1b-000000@eu-west-1.amazonaws.com designates 54.240.89.127 as permitted sender) smtp.mailfrom=010201966d20033b-afe7ec46-7db2-4b82-b241-4bca35c77f1b-000000@eu-west-1.amazonaws.com;
    dmarc=pass (p=QUARANTINE sp=QUARANTINE dis=NONE) header.from=instructure.com
Return-Path: <010201966d20033b-afe7ec46-7db2-4b82-b241-4bca35c77f1b-000000@eu-west-1.amazonaws.com>
Received: from a89-127.smtp-out.eu-west-1.amazonaws.com (a89-127.smtp-out.eu-west-1.amazonaws.com [54.240.89.127])
    by mx.google.com with ESMTPS id ffacdb85a97d-3a073e3afdc511204483f8f.202.2025.04.25.06.25.37
    for <nicodalessandro11@gmail.com>
    (version=TLS3_3 cipher=TLS_AES_128_GCM_SHA256 bits=128/128);
    Fri, 25 Apr 2025 06:25:37 -0700 (PDT)
Received-SPF: pass (google.com: domain of 010201966d20033b-afe7ec46-7db2-4b82-b241-4bca35c77f1b-000000@eu-west-1.amazonaws.com designates 54.240.89.127 as permitted sender) client-ip=54.240.89.127;
Authentication-Results: mx.google.com;
    dkim=pass header.i=@instructure.com header.s=hrt54zqnosd2q2fn3sy2vc6aumude2q header.b=HfF9WJ3M;
    dkim=pass header.i=@amazon.com header.s=lnchhvbugj3syuhssfvqohv723u4th header.b=UFA1uK6U;
    spf=pass (google.com: domain of 010201966d20033b-afe7ec46-7db2-4b82-b241-4bca35c77f1b-000000@eu-west-1.amazonaws.com designates 54.240.89.127 as permitted sender) smtp.mailfrom=010201966d20033b-afe7ec46-7db2-4b82-b241-4bca35c77f1b-000000@eu-west-1.amazonaws.com;
    dmarc=pass (p=QUARANTINE sp=QUARANTINE dis=NONE) header.from=instructure.com
DKIM-Signature: v=1; a=rsa-sha256; q=dns/txt; c=relaxed/smime; s=hrt54zqnosd2q2fn3sy2vc6aumude2q; d=instructure.com; t=1745587536; h=Date:From:Reply-To:To:Message-ID:Subject:Mime-Version:Content-Type:Content-Transfer-Encoding;
    bh=PhyKhfVCSaay8QSP0K1f1fJ0D9H/nJ2PK3jcvQq; b=HfF9WJ3Me25xfzKqnyRkC95dAyeV7XUa1721o4C1qU180f8eudBme1G0DV7y0 u5nKXaJAPKv182BcoJkda3D8PBKsP5a+7G2E0XblawQc5q2F7u3q8bBEPWw SQZrTtsr:UwPAlEt3+LdGnsLx09hg/vxqPM0h=
DKIM-Signature: v=1; a=rsa-sha256; q=dns/txt; c=relaxed/smime; s=lnchhvbugj3syuhssfvqohv723u4th; d=amazon.com; t=1745587536; h=Date:From:Reply-To:To:Message-ID:Subject:Mime-Version:Content-Type:Content-Transfer-Encoding:Feedback-ID;
    bh=PhyKhfVCSaay8QSP0K1f1fJ0D9H/nJ2PK3jcvQq; b=UFA1uK6U8f9EYvJBU194+13d4ahyV/kL5t/gfRS/hC2f8K8E/L00Z41C6GP /Jg2p2Ch8278WAsj9J0f8P2Q61QF2NylrId7XRFBtUuItPcAK25T1I1NkgM3T Eky/92uoneCF9pJpsu09PbebtYqVY26cq71RnoQ=
X-On-bounce-route-to: notification-service-failures-dub-prod
Date: Fri, 25 Apr 2025 13:25:36 +0000
From: Software Design Patterns - Classroom 1 <notifications@instructure.com>
Reply-To: reply+a83261586d5275b3-18667-77281625-1745587534@eu.notifications.canvaslms.com
To: nicodalessandro11@gmail.com
Message-ID: <010201966d20033b-afe7ec46-7db2-4b82-b241-4bca35c77f1b-000000@eu-west-1.amazonaws.com>
Subject: Solution PR1: Software Design Patterns - Classroom 1
Mime-Version: 1.0
Content-Type: multipart/alternative; boundary="---_mimepart_680b8d58ba88_746ac29cc13582"; charset=UTF-8
Content-Transfer-Encoding: 7bit
Auto-Submitted: auto-generated
Feedback-ID: ::1.eu-west-1.CDF8pFQ07SahQpZB7I47qGHf35OPZFHWZ169bek=:AmazonSES
X-SES-Outgoing: 2025.04.25-54.240.89.127
-----_mimepart_680b8d58ba88_746ac29cc13582
Content-Type: text/plain; charset=UTF-8
Content-Transfer-Encoding: quoted-printable

```


a. Identify and comment on the main fields that appear in the header.

- **Delivered-To:** Indicates the email receiver.
- **Received:** Shows when and through which servers the email has passed.
- **Return-Path:** The address that receives the answers if the delivery fails (bounce).
- **X-Google-Smtp-Source:** Internal identificatory of Google for tracking the e-mail.
- **Subject:** Subject of the e-mail.
- **From:** Email sender.
- **Message-ID:** Unique identifier of the e-mail.
- **MIME-Version:** Indicates that the message is in MIME format. Standard for sending non-ASCII files through emails.
- **Content-Type:** The type of content that the email carries, in this case text.
- **Date:** Date and time when the email was sent.

b. How many "Received" lines are in the header? What information do these lines provide?

We can identify three different "Received" lines in the header:

1. Google server reception with its specific ID.
2. A reception from the Amazon server SES in Europe.

Each line shows the path that the message follows from the origin server to its destination. This line includes information such as the server, the IP address, and the date.

Note: We observe also a SPF verification (that may be considered as a special type of reception) confirming the sender is authorized.

c. Are there any other lines that are repeated in the header? Which one is it and what information does it contain?

There are other lines that are repeated in the header such as the ones with the *ARC-* prefix, ARC-Message-Signature, ARC-Authentication-Results, ARC-Seal. There's other lines like DKIM-Signature, Return-Path, all of them associated with security mechanisms. DKIM verifies that the contents were not compromised. SPF for server sender confirmation as we already explained, and ARC that stores the history of server verifications, helping with authentication.

4. We want to send a file of size **F = 9 Gbits** to 5 peers. Suppose the server has an upload speed of **u = 85 Mbps**. The 5 peers have the following upload speeds:

- u1 = 14 Mbps
- u2 = 21 Mbps
- u3 = 15 Mbps
- u4 = 14 Mbps
- u5 = 17 Mbps

The 5 peers have the following download speeds:

- d1 = 28 Mbps
- d2 = 36 Mbps
- d3 = 23 Mbps
- d4 = 12 Mbps
- d5 = 37 Mbps

Note: You can find similar exercises at the following link:

https://gaia.cs.umass.edu/kurose_ross/interactive/CS_vs_P2P_download.php

- a. What is the minimum time required to distribute this file from the central server to the 5 peers using the client-server model?

In the client-server model, the server must send the entire file to each peer individually. So, to calculate the minimum distribution time T we will use the formula used in the example exercise.

$$T = \max\left(\frac{N \times F}{u_s}, \frac{F}{\min(d_1, d_2, d_3, d_4, d_5)}\right)$$

In our example:

$N = 5$ (number of peers)

$F = 9$ Gbits (file size)

$u_s = 85$ Mbps (server upload rate)

d_n = Download rate of each peer

$$T = \max \left(\frac{5 \times 9 \text{ Gbits}}{85 \text{ Mbps}}, \frac{9 \text{ Gbits}}{\min(28, 36, 23, 12, 37) \text{ Mbps}} \right)$$

$$T = \max \left(\frac{45000 \text{ Mbits}}{85 \text{ Mbps}}, \frac{9000 \text{ Mbits}}{12 \text{ Mbps}} \right)$$

$$T = \max (529.4 \text{ s}, 750 \text{ s})$$

The minimum time needed to distribute the file using the client/server model = 750 Seconds

- b. What is the minimum time required to distribute this file from the central server to the 5 peers using a peer-to-peer download?

In the peer-to-peer model, the server must send at least one copy of the file and the slower download rate will be still limiting. To calculate the minimum distribution time T we will use the formula used in the example exercise.

$$T = \max \left(\frac{F}{u_s}, \frac{F}{\min(d_1, d_2, d_3, d_4, d_5)}, \frac{N \times F}{u_s + \sum u_i} \right)$$

In our example:

$N = 5$ (number of peers)

$F = 9 \text{ Gbits}$ (file size)

$u_s = 85 \text{ Mbps}$ (server upload rate)

d_n = Download rate of each peer

$$T = \max \left(\frac{9000 \text{ Mbits}}{85 \text{ Mbps}}, \frac{9000 \text{ Mbits}}{12 \text{ Mbps}}, \frac{5 \times 9000 \text{ Mbits}}{85 + (14 + 21 + 15 + 14 + 17) \text{ Mbps}} \right)$$

$$T = \max (105.88 \text{ s}, 750 \text{ s}, 271.08 \text{ s})$$

The minimum time needed to distribute the file using the peer-to-peer model = 750 Seconds

We can see here that the system is again dominated by the slowest download rate of the peers (12 Mbps) which represents the system bottleneck.

5. Imagine you are designing a REST API for a task management application (to-do list). The application should allow users to create, read, update, and delete tasks. Each task has the following attributes:

- **id:** Unique task identifier.
- **title:** Title of the task.
- **description:** Detailed description of the task.
- **dueDate:** Due date of the task.
- **completed:** State of the task (true/false).

a. Create a table with the endpoints needed to perform CRUD (Create, Read, Update, Delete) operations on the tasks. Include the HTTP method, the path (endpoint), and a brief description of what each endpoint does.

We will create the NIA-API model with the required operations:

HTTP Method	CRUD	Endpoint	Description
POST	Create	/nia-api/tasks	To create a new task.
GET	Read	/nia-api/tasks	To obtain a list of all the task in the database.
GET	Read	/nia-api/tasks/{id}	To obtain a task given a specific id.
PUT	Update	/nia-api/tasks/{id}	To update a specific task given a specific id.
DELETE	Delete	/nia-api/tasks/{id}	To delete a specific task given a specific id.

- b. Provide an example of a JSON response when we retrieve a task by its ID. Add a version to the API (e.g., v1) and modify the endpoints to include the version.

We will first add a version V1 to our current API model NIA-API:

HTTP Method	CRUD	Endpoint	Description
POST	Create	/nia-api/v1/tasks	To create a new task.
GET	Read	/nia-api/v1/tasks	To obtain a list of all the task in the database.
GET	Read	/nia-api/v1/tasks/{id}	To obtain a task given a specific id.
PUT	Update	/nia-api/v1/tasks/{id}	To update a specific task given a specific id.
DELETE	Delete	/nia-api/v1/tasks/{id}	To delete a specific task given a specific id.

If we execute the request **GET /nia-api/v1/task/321**, we will obtain the following JSON response:

```
{
  "id": 321,
  "title": "Submission of assignment CA2",
  "description": "Networks and Internet Applications - CA2 Assignment: the reason why a computer network exists ",
  "dueDate": "2025-04-27",
  "completed": false
}
```

c. Specify which HTTP status codes each endpoint would return in case of success or error.

HTTP Method	Endpoint	Success Code	Error Code
POST	/nia-api/v1/tasks	201 Created	400 Bad request 500 Internal server error
GET	/nia-api/v1/tasks	200 OK	500 Internal server error
GET	/nia-api/v1/tasks/{id}	200 OK	404 not found 500 Internal server error
PUT	/nia-api/v1/tasks/{id}	200 OK	400 bad request 404 Not found 500 Internal server error
DELETE	/nia-api/v1/tasks/{id}	204 No content	404 Not found 500 Internal server error

- **201 Created:** Is used when a new task is created successfully.
- **204 No content** is used when a task is deleted successfully (no content to return).
- **404 Not found** is used when the task ID does not exist.
- **400 Bad Request** is used when the request is invalid, for example if there's a missing field.
- **500 Internal Server Error** is used when something goes wrong on the server.

6. YouTube uses a Content Delivery Network (CDN) to efficiently distribute its videos to users around the world.

a. Describe how content is replicated across different servers and delivered to the client.

When we upload a video to YouTube, the content is not being stored in a single place but replicated in many servers distributed alongside the world. These servers compose what is called the CDN or Content Delivery Network.

So, when a user is watching a video, YouTube first detects the user location by tracking the IP address, and instead of sending it to the root server that may be very far from the user location, connects the request with the closest server containing a copy of the video. This way, YouTube significantly improves the user experience, since the video quality will be better, it will load faster, have less pauses, etc.

b. Cite some examples of other popular CDNs.

Among others, these are the most popular CDNs nowadays:

- Google Cloud CDN
- Amazon CloudFront
- Microsoft Azure CDN
- Cloudflare
- Fastly

c. Open the Pingdom tool: <https://tools.pingdom.com>. Enter the URL of a website that uses a CDN (for example, <https://www.youtube.com>). Select three test locations (United States, Europe, and Asia) and click "Start Test" to measure the website's load time. Note the load time and the location of the server from which the content was served.

Pingdom Website Speed Test

Enter a URL to test the page load time, analyze it, and find bottlenecks.

URL


Test from

Asia - Japan - Tokyo

START TEST

The internet is fragile. Be the first to know when your site is in danger.

START YOUR FREE TRIAL



Your Results:

DOWNLOAD HAR

SHARE RESULT

	Performance grade A 92	Page size 175.0 KB
	Load time 317 ms	Requests 13

Pingdom Website Speed Test

Enter a URL to test the page load time, analyze it, and find bottlenecks.

URL

Test from Europe - United Kingdom - London

[START TEST](#)

The internet is fragile. Be the first to know when your site is in danger.

[START YOUR FREE TRIAL](#)

Your Results:

[DOWNLOAD HAR](#) [SHARE RESULT](#)

	Performance grade	Page size
	A 92	174.4 KB
	Load time	Requests
	217 ms	13

Pingdom Website Speed Test

Enter a URL to test the page load time, analyze it, and find bottlenecks.

URL

Test from North America - USA - San Francisco

[START TEST](#)

The internet is fragile. Be the first to know when your site is in danger.

[START YOUR FREE TRIAL](#)

Your Results:

[DOWNLOAD HAR](#) [SHARE RESULT](#)

	Performance grade	Page size
	A 92	175.0 KB
	Load time	Requests
	291 ms	13

- d. Enter the URL of a website that does not use a CDN (for example, a small blog or a local site). Run the previous test in the same locations as in the previous step. Note the load time and server location and comment on the results.

Pingdom Website Speed Test

Enter a URL to test the page load time, analyze it, and find bottlenecks.

URL


Test from

Asia - Japan - Tokyo


START TEST

The internet is fragile. Be the first to know when your site is in danger.

START YOUR FREE TRIAL



Your Results:



Performance grade

D 67

Page size

1.1 MB

Load time

4.00 s

Requests

66

DOWNLOAD HAR

SHARE RESULT

Pingdom Website Speed Test

Enter a URL to test the page load time, analyze it, and find bottlenecks.

URL


Test from

Europe - United Kingdom - London


START TEST

The internet is fragile. Be the first to know when your site is in danger.

START YOUR FREE TRIAL



Your Results:



Performance grade

D 70

Page size

3.4 MB

Load time

1.56 s

Requests

111

DOWNLOAD HAR

SHARE RESULT

Pingdom Website Speed Test

Enter a URL to test the page load time, analyze it, and find bottlenecks.


URL
 https://beteve.cat/

Test from
 North America - USA - Washington D.C

START TEST


The internet is fragile. Be the first to know when your site is in danger.

START YOUR FREE TRIAL



Your Results:

DOWNLOAD HAR
 SHARE RESULT



Performance grade
D 67

Page size
4.2 MB

Load time
2.43 s

Requests
157

Conclusion:

1. After running the different test proposed on the **YouTube** page we obtain:

Location	Load Time	Page Size	Requests
Asia (Tokyo)	317 ms.	175 KB	13
Europe (London)	217 ms.	174.4 KB	13
North America (San Francisco)	291 ms.	175 KB	13

We can observe that all load times are under 400 ms. across the world. The page size is very small, and we have very few requests. This shows the power of CDN and how it works perfectly worldwide.

2. After running the different test proposed on the **Betevé**, a non CDN site we obtain:

Location	Load Time	Page Size	Requests
Asia (Tokyo)	4 seconds	1.1 MB	66
Europe (London)	1.56 seconds	3.4 MB	111
North America (San Francisco)	2.43 seconds	4.2 MB	157

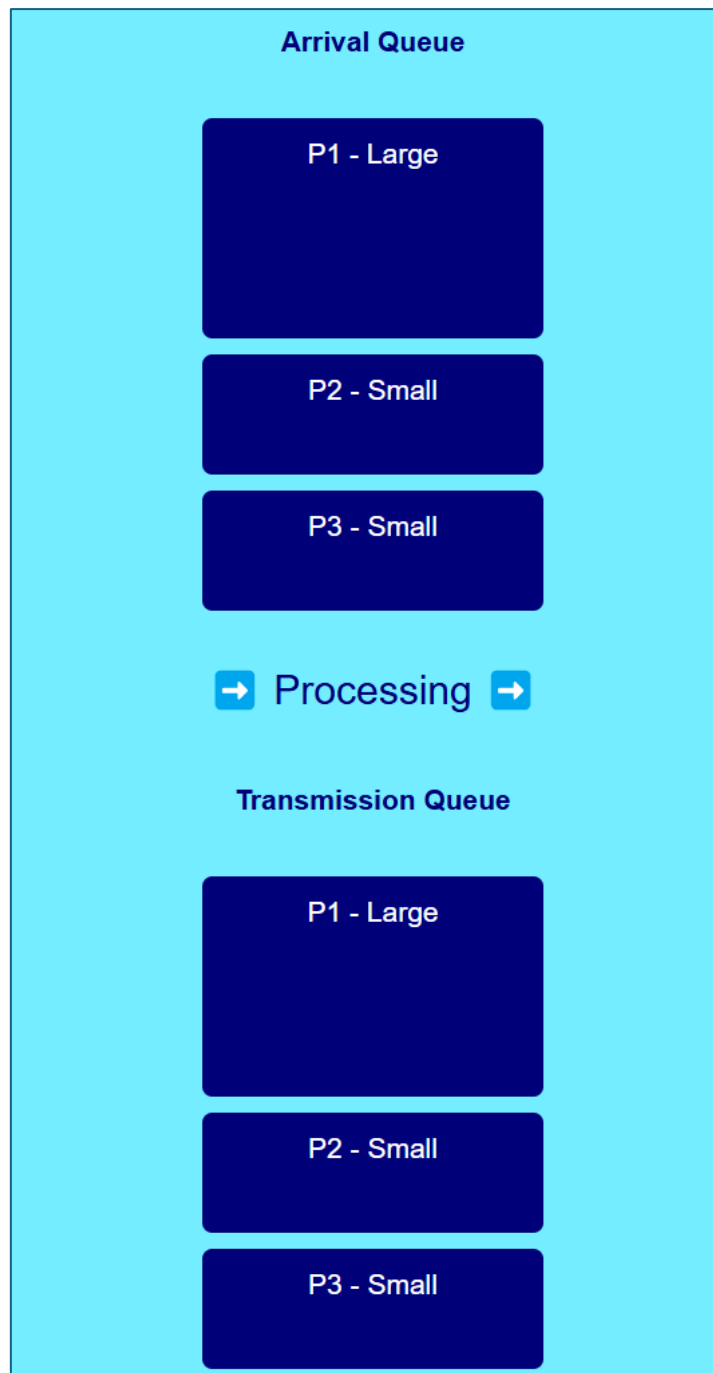
In this example the load times are much higher (in Asia we have almost 4 seconds). The page size is much larger than the YouTube example (up to 4.2 MB), and we have many more requests. This illustrates that users farther from the server will have a much worse experience.

18

7. Regarding packet scheduling algorithms, provide reasoned answers to the following questions:

- a. In First Come First Served (FCFS), what happens if a large packet arrives first? How does this affect the waiting time for smaller packets?

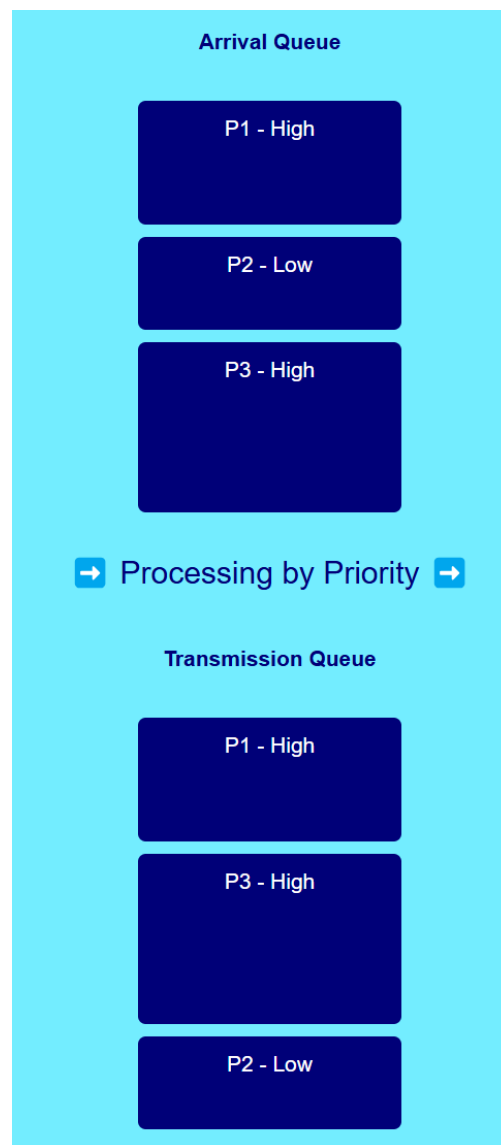
If a large packet arrives first, it will be processed entirely before the rest of the packages. This is called “head-of-line” blocking and is considered to have various negative effects. When this packet arrives, it occupies the link for a longer time causing that the smaller packets have longer wait times.



- b. In Priority Scheduling, suppose three packets arrive in this order: P1 (high priority, size 100), P2 (low priority, size 50), P3 (high priority, size 200). What happens in this case if a low-priority packet arrives before a high-priority packet? Is the priority respected in the simulation? What is the transmission order of the packets?

In Priority Scheduling, the packets are transmitted based on their priority (no matter the arrival order). In this model, priority is respected meaning that high priority packages are transmitted first, even if a low priority packet arrived earlier. In our example it will be:

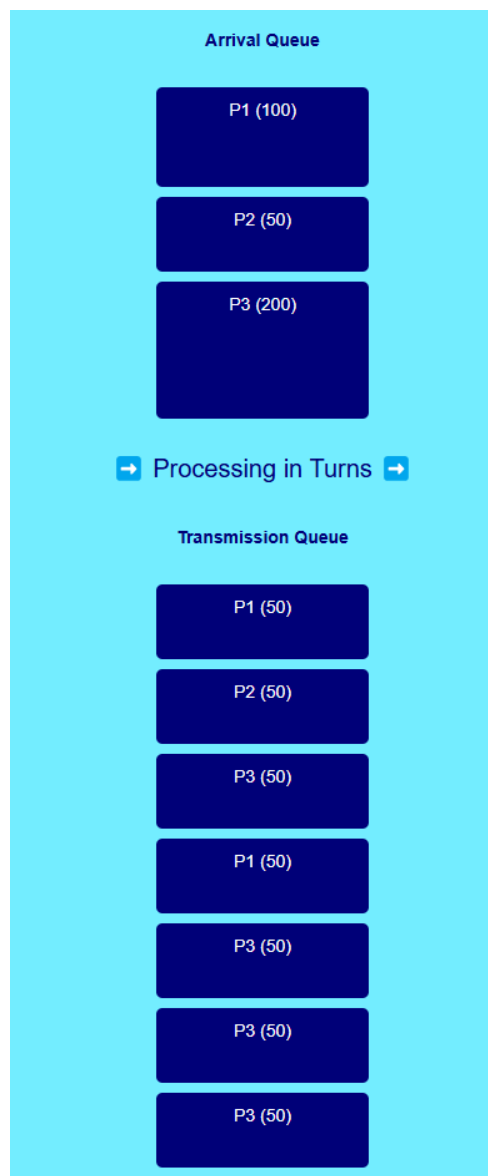
- P1 (high priority, size 100) is transmitted first
- P3 (high priority, size 200) is transmitted second
- P2 (low priority, size 50) is transmitted last



- c. In Round Robin, suppose we want to transmit three packets: P1 (size 100), P2 (size 50), P3 (size 200). The quantum is 50. What happens if the packets have very different sizes? What will the order of the packets be?

In this case, given that the quantum is 50, the packets will alternate in turns of 50 units. So, smaller packets complete first, and the larger packets will require multiple rounds. In our example the order will be:

- P1 send 50 units, wait.
- P2 sends 50 units (completed).
- P3 send 50 units, wait.
- P1 sends 50 units (completed).
- P3 sends 3 more rounds of 50 units each (completed).



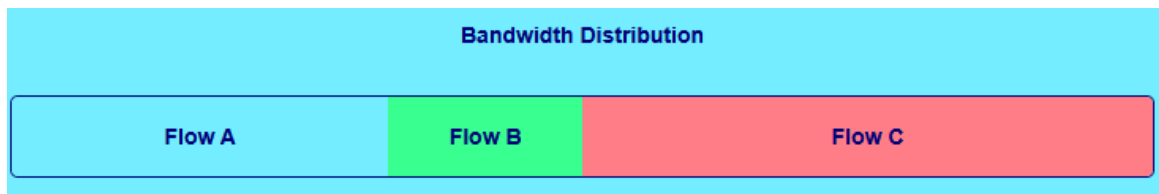
d. In Weighted Fair Queueing (WFQ), suppose we have three flows with the following weights:

- I. Flow A: Weight 2
- II. Flow B: Weight 1
- III. Flow C: Weight 3

How will the bandwidth be distributed among the different flows?

In this case, the bandwidth is divided proportionally based on the weights of the flows. In our example:

- Flow A: Weight 2 – 33.3%
- Flow B: Weight 1 – 16.7%
- Flow C: Weight 3 – 50%



8. Regarding HTTP protocol:

a. Perform a comparative analysis of the HTTP/1.1, HTTP/2, and HTTP/3 protocol versions, paying attention to the following characteristics:

- Year of release
- Current adoption level
- Main advantages
- Main limitations

Feature	HTTP/1.1	HTTP/2	HTTP/3
Year of release	1997 (revised in 1999)	2015	2022
Current adoption level	★★★★ Very High Widely supported, still in use, especially in legacy systems.	★★★★ High High adoption among majority of websites and CDN.	★★★★ High Growing adoption, supported major browsers and CDN
Main advantages	Easy to use and fix. Simple and widely understood. Persistent connections via keep/alive.	Binary framing for efficiency. Prioritization of streams and Multiplexing (multiple streams over one connection).	Works better on phones and also when the internet connections is not optimal. Connects faster: Built on QUIC (UDP based) which is faster for the handshakes. No head of line blocking at transport level. Sends many things at once. Servers can send files first.
Main limitations	Head of line blocking the application layer. One request per connection Redundant headers in each request	Still subject to TCP level head of line blocking. More complex implementation.	Depends on UDP and it may be blocked in some enterprise networks. Is new and is still evolving Ecosystem still maturing.

b. What do you think will be the short- and medium-term future trends for the three versions?

- In the short term, I think that HTTP/1.1 will continue to be supported in legacy systems, but its adoption will decline in favor of the newer versions. HTTP/2 will be the dominant because of its compatibility, maturity and performance improvements. Finally, HTTP/3 will increase its adoption given its low latency, especially in mobile and video applications.
- In the medium term, HTTP/1.1 will become almost obsolete for modern applications, and will only exist in specific applications or legacy systems. HTTP/2 will be gradually be replaced by HTTP/3 specially designed for high performance environments but will still serve as option where UDP is unsupported. HTTP/3 likely become the default standard for most of the web applications, especially where speed and user experience are critical.

9. GraphQL (<https://graphql.org/>) is an API query language developed by Facebook in 2012 and released as open-source in 2015. Research this technology and answer the following questions:

a. What main advantages does GraphQL offer over REST?

1. With this technology you can request only what you need. In this query language, the client defines what is the data needed.
2. One request instead of many. With this technology you can obtain multiple resources with only one query.
3. Better for slow connections.
4. More flexibility for the frontend since you can control the response and adapt it to the different user views and devices.
5. No versioning problems since you can add new fields without breaking existing apps.

b. What disadvantages or limitations does GraphQL have compared to REST?

1. While REST has been the most common and widely adopted language for API development in the last years, adopting GraphQL will add complexity for the initial setup, since it requires learning a whole new query language that involves designing schemas and resolvers.
2. Cache control becomes harder given that in REST you can easily adopt the HTTP cache system (URL based caching), but in GraphQL this task can be more varied.
3. In contrast with REST, when using GraphQL we may have very complex queries that can overload the servers resulting in very large and expensive resource consumption.
4. Debug can be more complicated and difficult to track in comparison with REST, especially when you have nested resolvers.

c. What are queries and mutations in GraphQL?

1. Queries are the equivalent to GET in rest. They are used to request and retrieve data.
2. Mutations are the equivalent to POST, PUT, and DELETE in REST. They are used to create, update or delete data.

d. Suppose you are implementing an API to manage your favorite movies using GraphQL. Each movie has the following attributes:

- **id:** Unique identifier of the movie.
- **title:** Title of the movie.
- **year:** Release year.
- **genre:** Genre of the movie.

You want to perform the following operations:

- Queries:
 - Get a list of all movies.
 - Get a specific movie by its ID.
- Mutations:
 - Add a new movie.

I. Define the data types for Movie.

Following the above description, we can define the following Movie type:

```
type Movie {
  id: ID!
  title: String!
  year: Int
  genre: String
}
```

II. Define the necessary queries and mutations.

We will first define a query that can get all the movies and one movie by its id:

```
type Query {
  movies: [Movie]!
  movie(id: ID!): Movie
}
```

Now, we define a Mutation that can create a new movie with the given information, returning a new created movie:

```
type Mutation {
  addMovie(title: String!, year: Int!, genre: String!): Movie!
}
```

10. Search information about two different types of cloud computing models, one for IaaS, PaaS or SaaS and the other for FaaS (Function as a Service) or BaaS (Backend as a Service).

a. Briefly describe the main features of each model.

- **Model 1:** The **SaaS** model are applications that can be used directly from the internet without the need of installing it in your local machine. The provider is in charge of managing the infrastructure such as maintenance, security, updates, etc. It usually has a subscription model (monthly or yearly) and the data is stored in the cloud.
- **Model 2:** The **BaaS** model are services already made for backend functionalities such as user authentication, databases, cloud storage, etc. It is like “prefabricated parts” for our app, so we don’t have to build everything from scratch. You can easily connect the app with these services saving a lot of development time.

b. List two use cases that are suitable for the first model selected and two that are suitable for the second model selected.

- For the **SaaS** model, many back-office services have nowadays adopted this model. In my case, I am currently using at work Asana for project management and Microsoft 365 for online collaboration. The company is also adopting Salesforce which is a CRM service that is under this model.
- Regarding the **BaaS** model I will mention the use case when you need to launch an MVP quickly when building applications, where services like Firebase (very popular for mobile apps) or Supabase allow to setup your application very easy without a backend setup, offering services like database, object storage (buckets), etc. Another use case can be for building mobile applications that require user login or real-time database notifications like a chat app, etc.

c. Find two companies that offer the second model selected services.

As mentioned, Firebase from Google, and Supabase are two clear examples of companies that offer BaaS services. We can also mention Backendless famous for its visual API (no code needed) and push notifications. Kuzzle, focus on real-time databases for IoT, geospatial analytics, etc.

d. Focusing on FaaS, it has some limitations, some of which are listed below. Describe these limitations and how they could be resolved:

- I. Limited execution time
- II. Cold start
- III. Vendor lock-in
- IV. Difficulty debugging and monitoring

Limitation	Description	How to resolve it
Execution time	You can run the service for a limited amount of time. (example Lambda 15 minutes limited time)	Split the task in small pieces or migrate to other services.
Cold Start	The first time that you run the service is being executed it may take more time since it has to build the service.	Prioritize the usage of languages that starts quickly like Go or Node.js. In AWS you can also keep the function active or used "provisioned concurrency".
Vendor lock-in	You are very tied to the provider because its unique technology characteristics difficulting its migration. (example Lambda function is different than Azure functions)	Use multiplatform tools or write code as standard as you can. Implement adapter patterns, etc.
Difficulty debugging and monitoring	It is difficult to debug errors and understand what is happening since you don't have access to the complete system.	Use other services for logging and monitoring such as AWS CloudWatch or Google Cloud logs. Test your functions in a local environment first.

NOTE: Based on my personal experience, I can illustrate how this FaaS constrains affect my daily routine. When developing ETL scripts at work, we should always prioritize based on costs, the use of AWS Lambda, which is a pure FaaS service. Even this service is very useful for most of the use cases, the above-described limitations (cold start, difficult to debug with cloud watch, and mainly the limited execution time when working with big data *15 minutes), we are forced to migrate to services like Glue or ECS that are services in the "middle ground" of FaaS and PaaS and have a different pricing model.

References

Kurose, James F. **Computer Networking: A Top-Down Approach**. Eighth edition. Boston [etc.] : Addison-Wesley, cop. 2021. ISBN 9781292405513

In addition to the learning material given for this unit, I have consulted the following materials in the internet:

Betevé.

"Portal de noticias y actualidad de Barcelona"

<https://beteve.cat/>

BitDegree.

"Explicación sobre algoritmos en informática"

<https://es.bitdegree.org/crypto/aprender/diccionario-crypto/que-es-el-algoritmo-fcfs>

IBM Documentation.

"Listado de mensajes de error en APIs de IBM"

<https://www.ibm.com/docs/es/qradar-common?topic=versions-api-error-messages>

Wikipedia.

"Artículo sobre computación en la nube"

https://es.wikipedia.org/wiki/Computaci%C3%B3n_en_la_nube

IBM Think Blog.

"Artículo explicativo sobre los modelos de servicio en la nube"

<https://www.ibm.com/es-es/think/topics/iaas-paas-saas>

AWS Lambda (Amazon Web Services).

"Documentación oficial sobre AWS Lambda y el modelo Function as a Service (FaaS)."

<https://aws.amazon.com/es/lambda/>