

Arjola Cara
baltt7

1. Compare the two computing models mainframe and cloud. Why did it change over time so dramatically?

Mainframe

Architecture	Centralized computing power; terminals connect to a single mainframe
Cost	Very expensive to purchase, maintain, and operate
Scalability	Limited and vertical (you upgrade the whole system)
Accessibility	Restricted to on-premises or private networks
Flexibility	Rigid; changes and upgrades are slow and complex
Use Cases	Banking, government, and large enterprises needing extreme reliability and processing power

Cloud

Architecture	Distributed computing via internet-connected data centers
Cost	Pay-as-you-go model; no large upfront costs
Scalability	Highly scalable, both vertically and horizontally
Accessibility	Accessible from anywhere via the internet
Flexibility	Very flexible; services and resources can be adjusted in real-time
Use Cases	Startups, enterprises, mobile apps, AI/ML workloads, data analytics, etc.

Cost Efficiency

Cloud computing eliminated the need for huge capital investments in hardware and staff. You only pay for what you use.

Scalability Needs

Businesses need to scale fast—cloud services like AWS, Azure, and Google Cloud can handle millions of users instantly.

Internet & Mobility

The rise of the internet and mobile devices demanded flexible, globally accessible computing—not possible with centralized mainframes.

Agility & Innovation

Cloud allows fast deployment, testing, and scaling of applications, which fuels innovation (especially in startups and agile companies).

Maintenance Burden

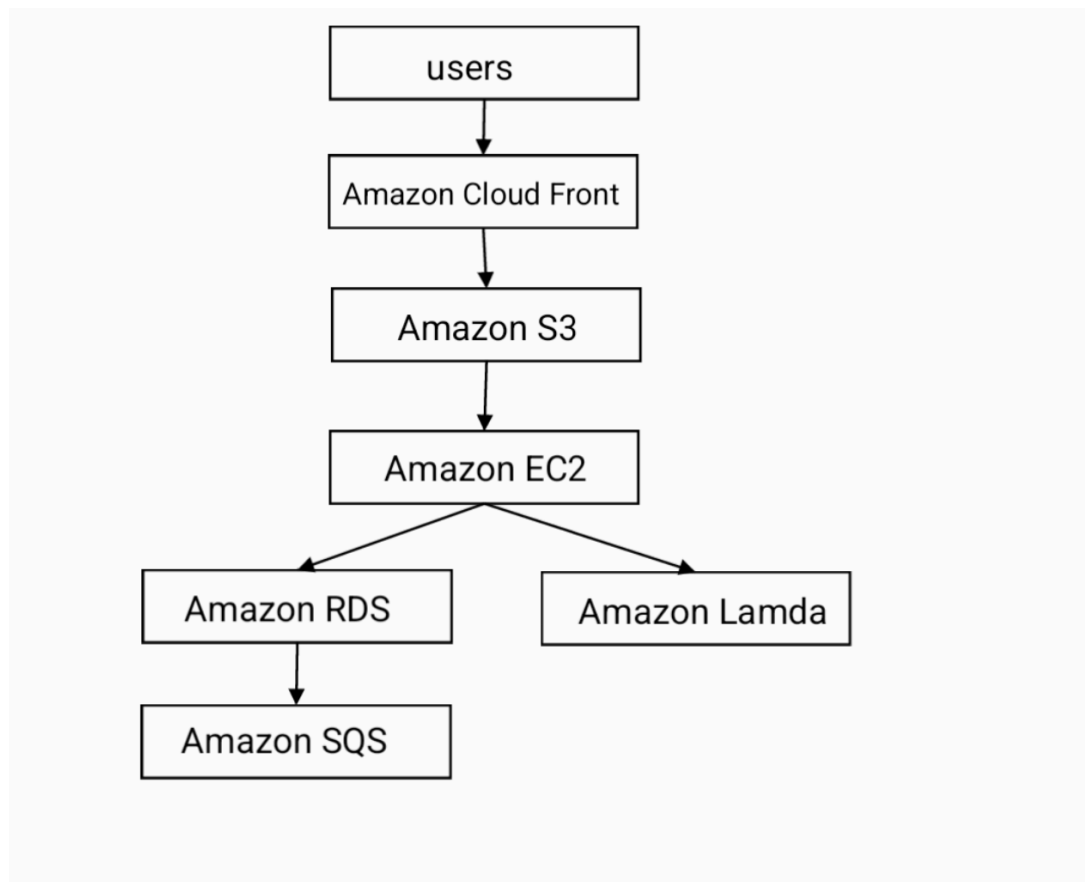
Mainframes require skilled personnel and costly maintenance; the cloud offloads that responsibility to the provider.

Globalization & Remote Work

Cloud services enable collaboration and access from anywhere—essential in today's remote and global working environments.

Task3

1. **Amazon EC2 (Elastic Compute Cloud):**
Use EC2 to run the main parts of your app. It can grow or shrink depending on how many users there are.
2. **Amazon S3 (Simple Storage Service):**
Store your app's data, backups, and files like images or videos in S3. It's big, safe, and easy to use.
3. **Amazon RDS (Relational Database Service):**
Use RDS to manage your database. It can handle more users when needed and keeps your data safe and available.
4. **Amazon SQS (Simple Queue Service):**
Use SQS to help different parts of your app talk to each other. It passes messages between them so they work better together.
5. **Amazon Lambda:**
Run small pieces of code with Lambda without having to manage servers. It's good for background tasks.
6. **Amazon CloudFront:**
Use CloudFront to show your app's content (like images or videos) quickly to users all over the world.



Task4

"Write a Python function that takes a list of numbers and returns the index of the first even number."

```
def find_first_even(numbers):  
    for i in range(len(numbers)):  
        if numbers[i] % 2 == 0:  
            return i  
    return -1
```

Wenn **negative Zahlen** oder **Gleitkommazahlen** (floats) in der Liste vorkommen, **funktioniert der Code nicht robust**.

```
print(find_first_even([3.0, 5.0, 6.0])) # gibt 2 zurück – aber 6.0 ist kein echter Integer  
print(find_first_even(["2", 4]))      # TypeError  
find_first_even(["3", "4", 5])  
# => TypeError: not all arguments converted during string formatting
```

```
def find_first_even(numbers):  
    for i, n in enumerate(numbers):  
        if isinstance(n, int) and n % 2 == 0:  
            return i  
    return -1
```