# Workloads in the cloud

- **Types of workloads**

- **Packaging options**

- **Execution options**

# New architectural concepts…
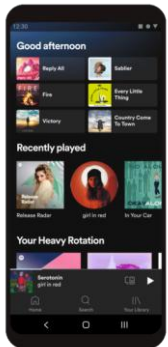
- **I want to motivate two design concepts…**

  1. **Lambda functions**

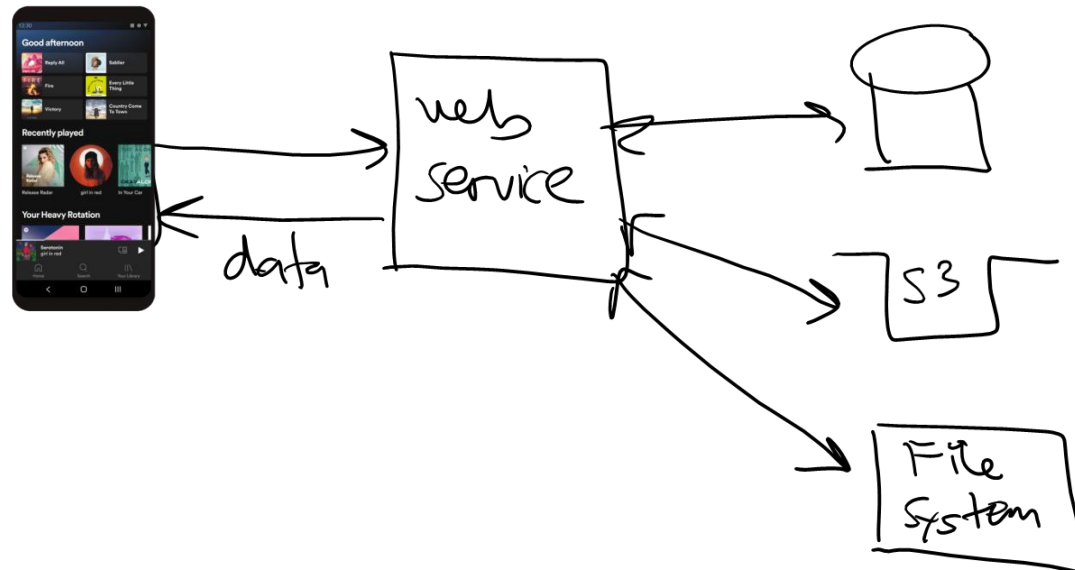  2. **Serverless computing**

# Multi-tier, data-driven apps

- **Our examples (so far) have all been data-driven**
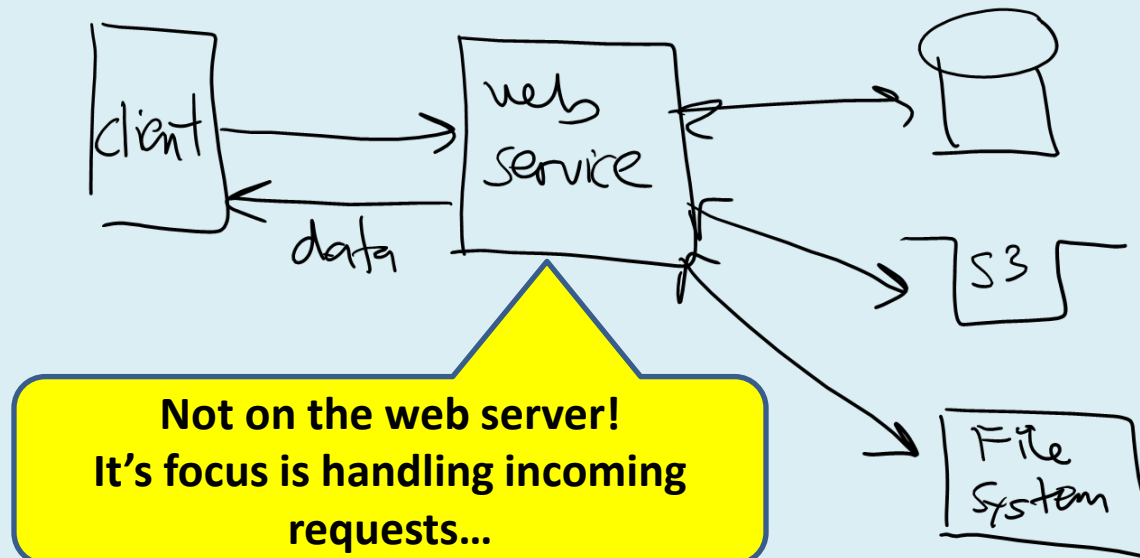
# I/O bound

- **We call this kind of workload "I/O bound"**

  – *Server is spending most of its time waiting for requests / data, i.e. input/output*

  – *This is typically handled via async programming*

# Compute-bound

- ## What if we need to compute something?
  - *Image/video compression, encryption, content analysis*
  - *Stock market simulation*
  - *Run AI / ML training set*

- ## We call these "compute-bound" workloads due to heavy CPU usage... Where do we execute?

**Not on the web server!**
**It's focus is handling incoming requests...**

# Example: prime factors in Python

https://2noicxltxjwxxt4ego5d7q4uc40bcgjw.lambda-url.us-east-2.on.aws/?n=600851475143

https://2noicxltxjwxxt4ego5d7q4uc40bcgjw.lambda-url.us-east-2.on.aws/?n=6008514751439999

```python
import json

def prime_factors(n):
  i = 2

  factors = []
  while i * i <= n:
    if n % i:
      i += 1
    else:
      n //= i
      factors.append(i)

  if n > 1:
    factors.append(n)

  return {
    'statusCode': 200,
    'body': json.dumps(factors)
  }
```
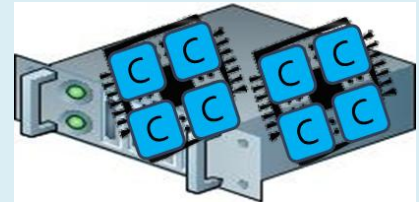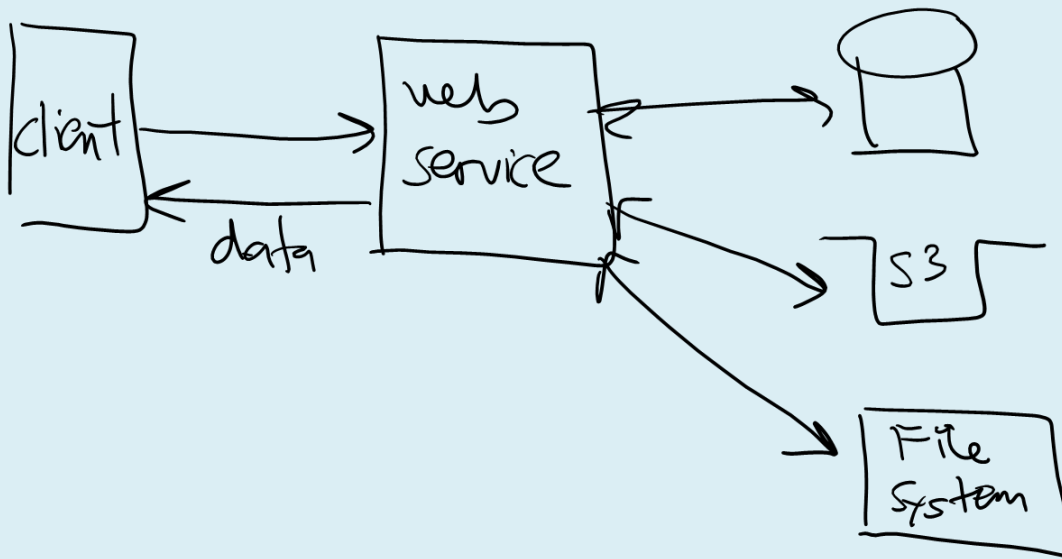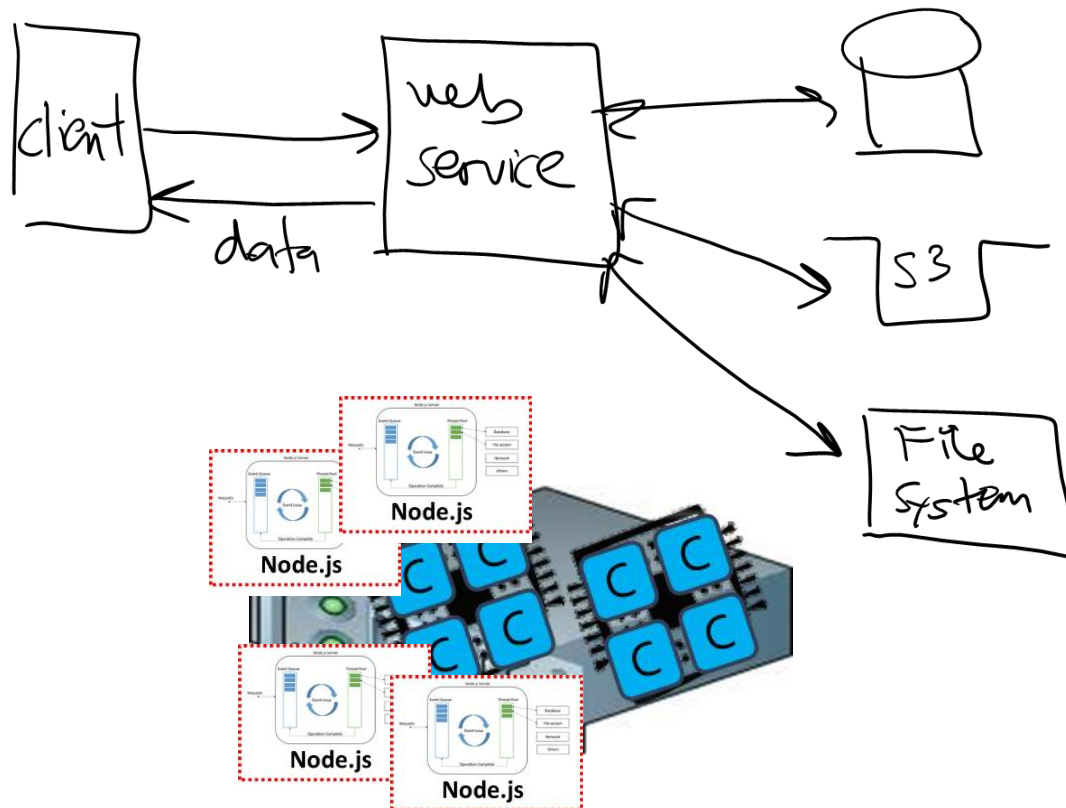


7

# Compute tier

- **We need a separate tier for executing compute-bound work**
  - *This can be a separate core, CPU, or machine*

# Option #1

- **\*IF\* you have unused cores available, use those**
  - *Better for small-scale work, i.e. small tasks that only run for a few seconds / minutes*
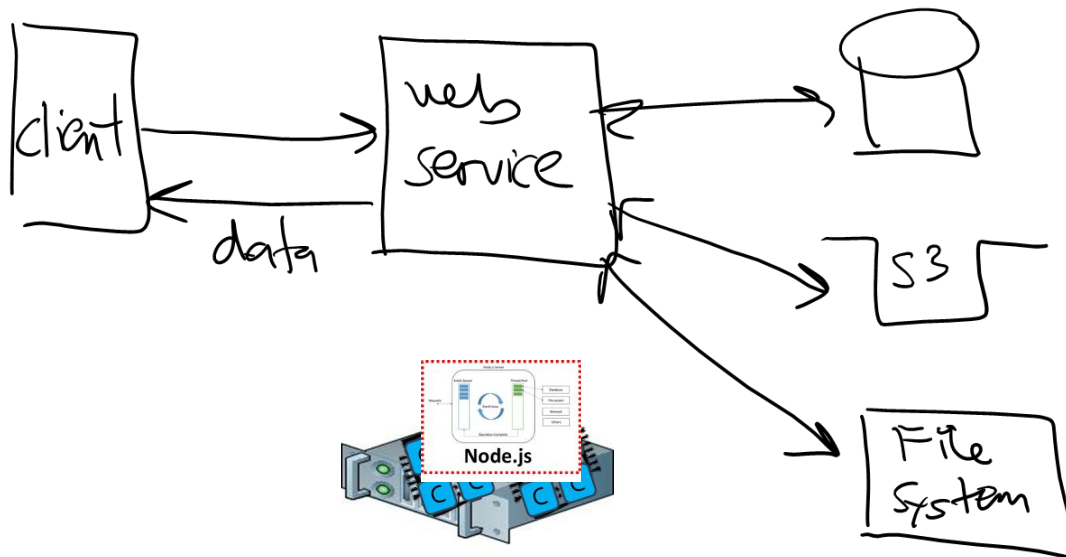
# Option #2

- **What if my task takes longer to run, or needs lots of RAM?**

- **Run on separate hardware…**



The problem? Installing the software you want to run…
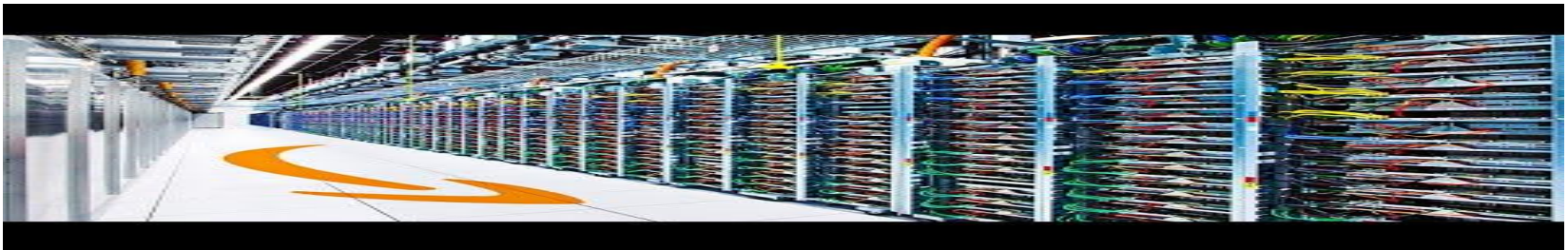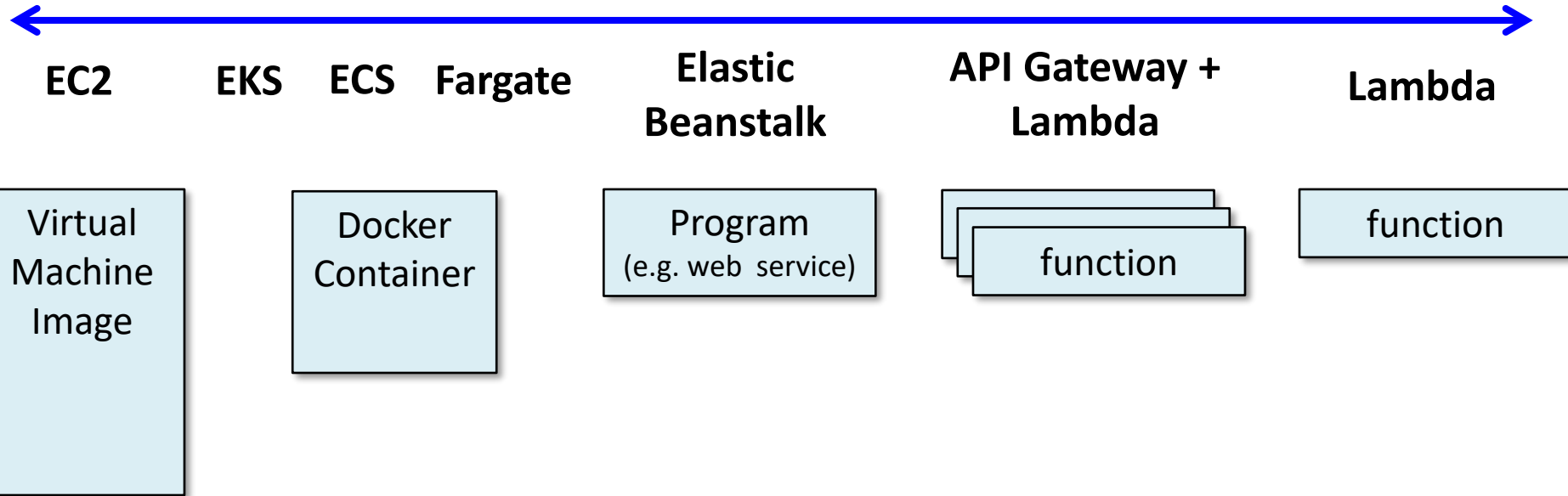
# Elastic Compute Cloud (EC2)

**Everything runs on an EC2 instance…**

**EC2 is AWS machine rental service, started in 2006**

• Outsourcing hardware is an old idea. Amazon's innovation was to charge by the **hour**, not month, and this started cloud revolution
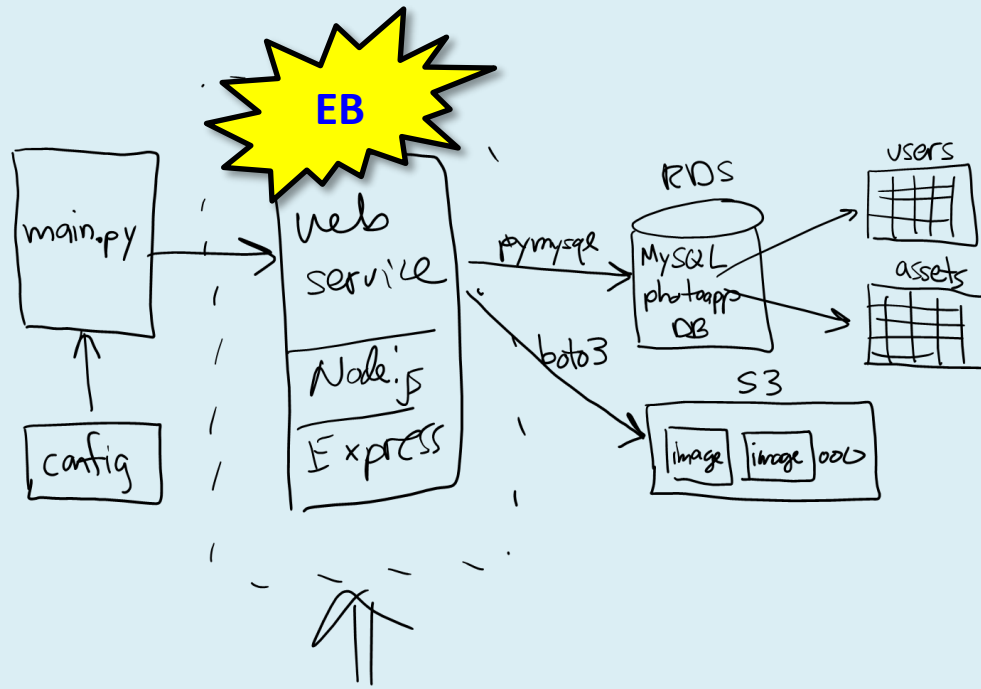
# Software packaging options

EC2    EKS    ECS    Fargate    Elastic Beanstalk    API Gateway + Lambda    Lambda

| Virtual Machine Image | Docker Container | Program (e.g. web service) | function | function |

# Example

Elastic Beanstalk

Program
(e.g. web service)

- **In project 02, EB allowed us to have a web service up and running with .zip and a few button clicks...**

# Execution continuum

- **Trade-offs:**

← ─────────────────────────────────────────── →

| **EC2, EKS, ECS, Fargate** | **Elastic Beanstalk** | **API Gateway + Lambda** | **Lambda** |

- *Run any software you want for as long as you want*
- *Complete control over HW and SW*
- *Hardest to config*

- *Server-based*
- *Upload .zip file*
- *Limited software choices*
- *Some control over HW and SW*

- *Function based*
- *Near-zero config*
- *Multi-tier web service + functions (15-min limit)*

- *Function based*
- *Near-zero config*
- *Short execution (< 15 mins)*

# That's it, thank you!