On the quest for

Elasticity

An architectural journey from servers to serverless computing

Scalability

The ability of a system to gracefully handle larger loads by adding computational resources.

- **Horizontal**: when more computing nodes are added;
- **Vertical**: when more hardware is added;
- **Diagonal**: combination of the above.

Elasticity

The ability to provision (and de-provision) computing resources, so that at each point in time the available resources match the current demand as closely as possible.

The beginning of a dangerous quest

- The (totally **not** evil) company you work for came up with a great idea: a website to send customizable, animated greeting cards via email. They'll call it «***Greetic***».
- They plan to profit from selling premium content, and to gather a little personal user data along the way.
- Your job is to make this happen as smoothly as possible, and so your journey begins…

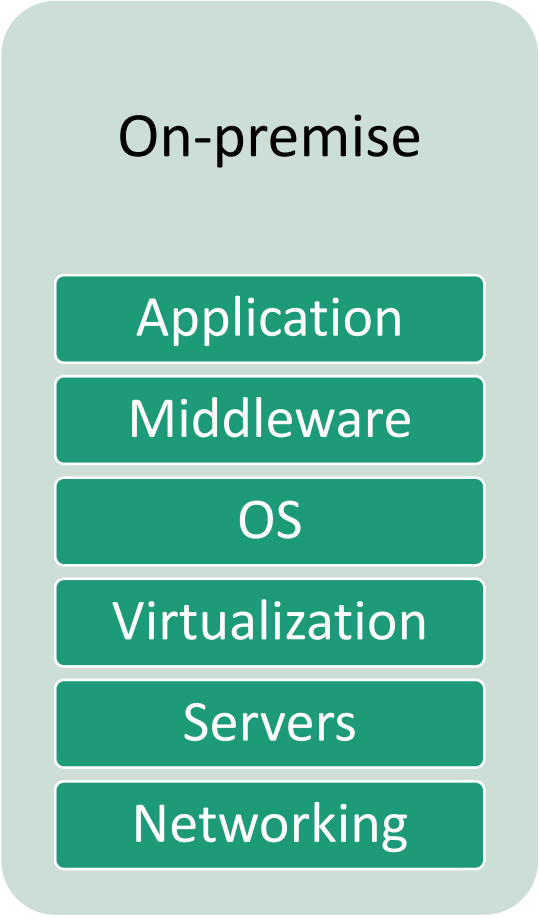
Emanuela Fanelli. *Una Pezza di Lundini*, RAI 2, 2021.

Chapter 1
The land of the **on-premise** forerunners

Pros and cons of self-hosted on-premise solutions.

The on-premise approach

- You manage the entire computing infrastructure, from the data center to the software.
- Pros:
  - Completely in control over own data and processes;
  - Reduced dependence from external factors.



(Partial) On-premise to-do list

- ❑ Estimate number of customers and computing needs;
- ❑ Buy servers (they’re expensive!);
- ❑ Find an adequate server room (that’s expensive!);
- ❑ Disaster recovery (that’s expensive!);
- ❑ Hire a System Administrator (that’s expensive!);

You have to pay quite a lot of money upfront, before you even have the chance to start making some dough!

Riddles for the adventurers

Is the on-premise solution **scalable**?

- Well, we can't really tell. That depends on how the greeting cards website is designed!

Is the on-premise solution **elastic**?

- **Not at all!**

We did all that and we're online. But…

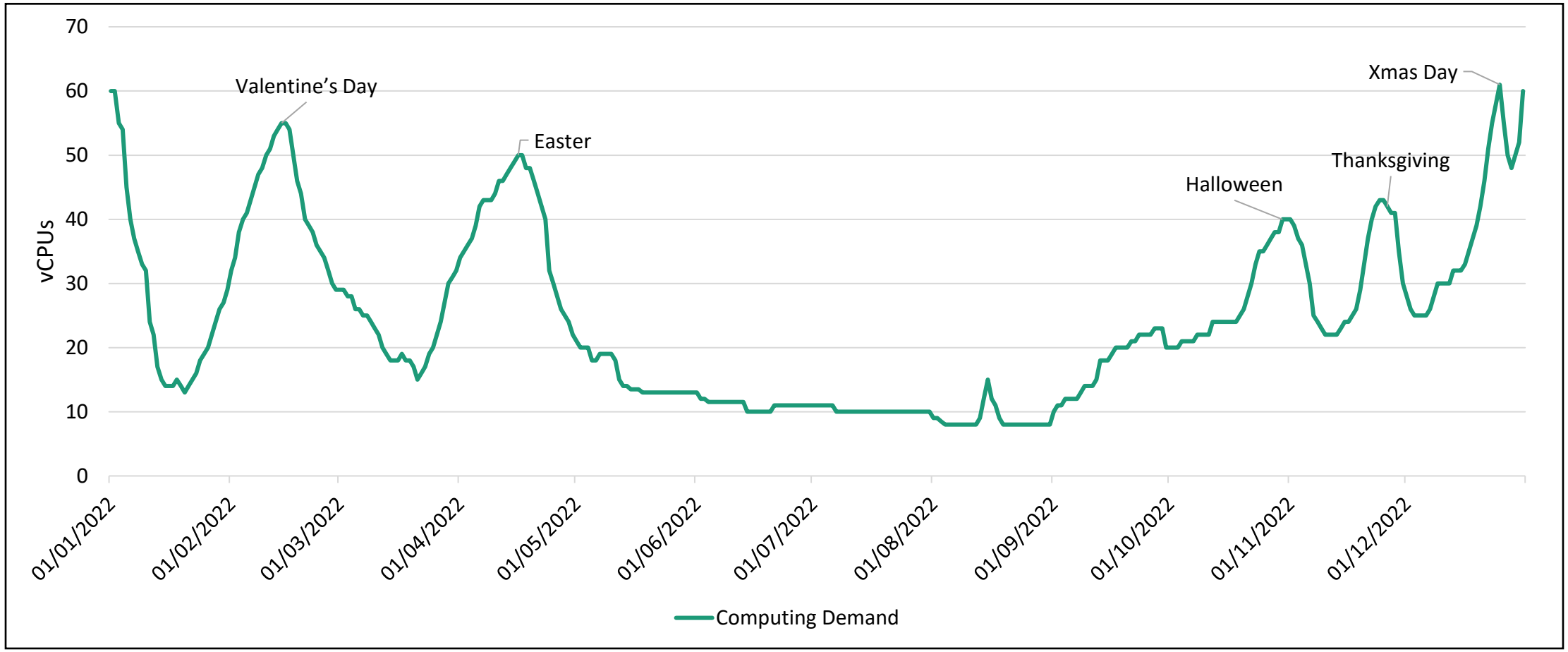
- What if we **overestimated** the number of customers?
- What if we **underestimated** the number of customers?

The dragon of unbalanced workloads

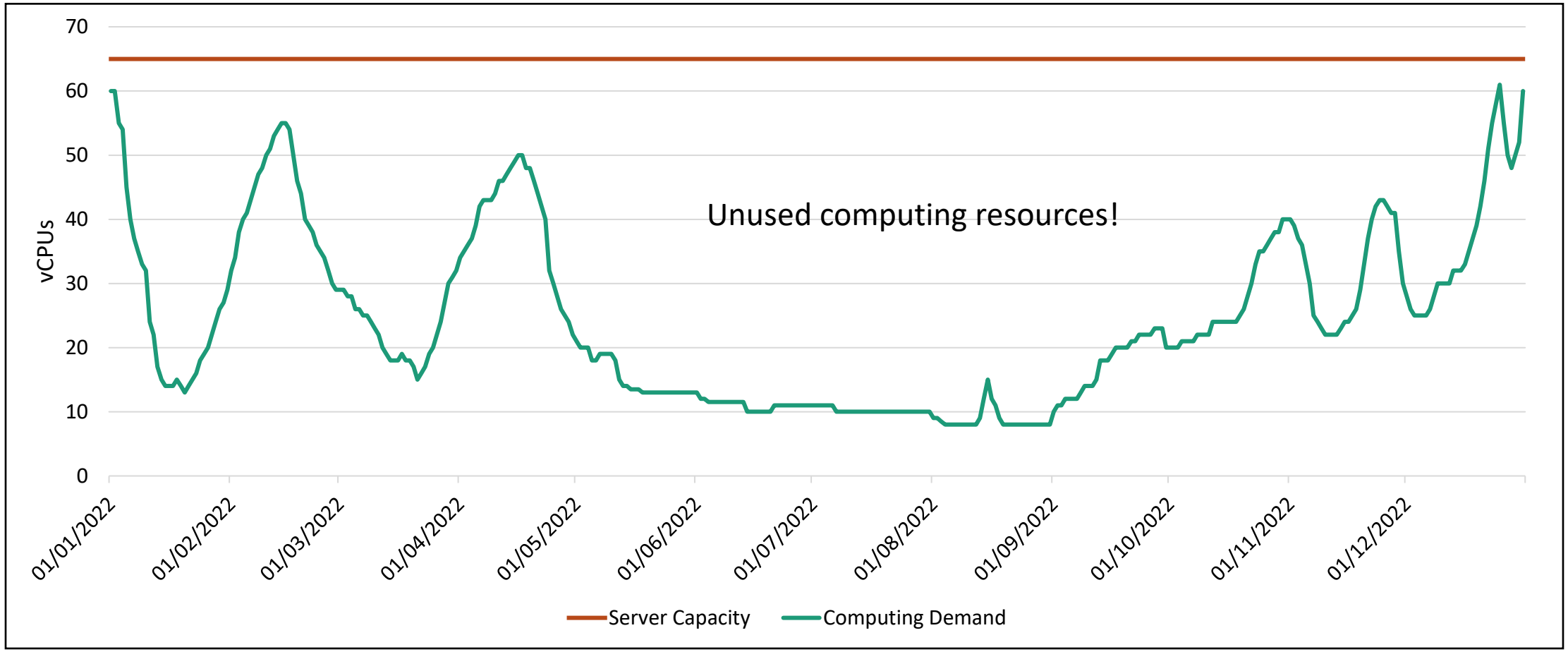
A greeting card generation website is **not** likely to have an uniform workload throughout the year.

- There will be usage spikes around the holidays…
- … and relatively low usage throughout the rest of the year.

Unbalanced workload for your system



Handling unbalanced workloads on-premise

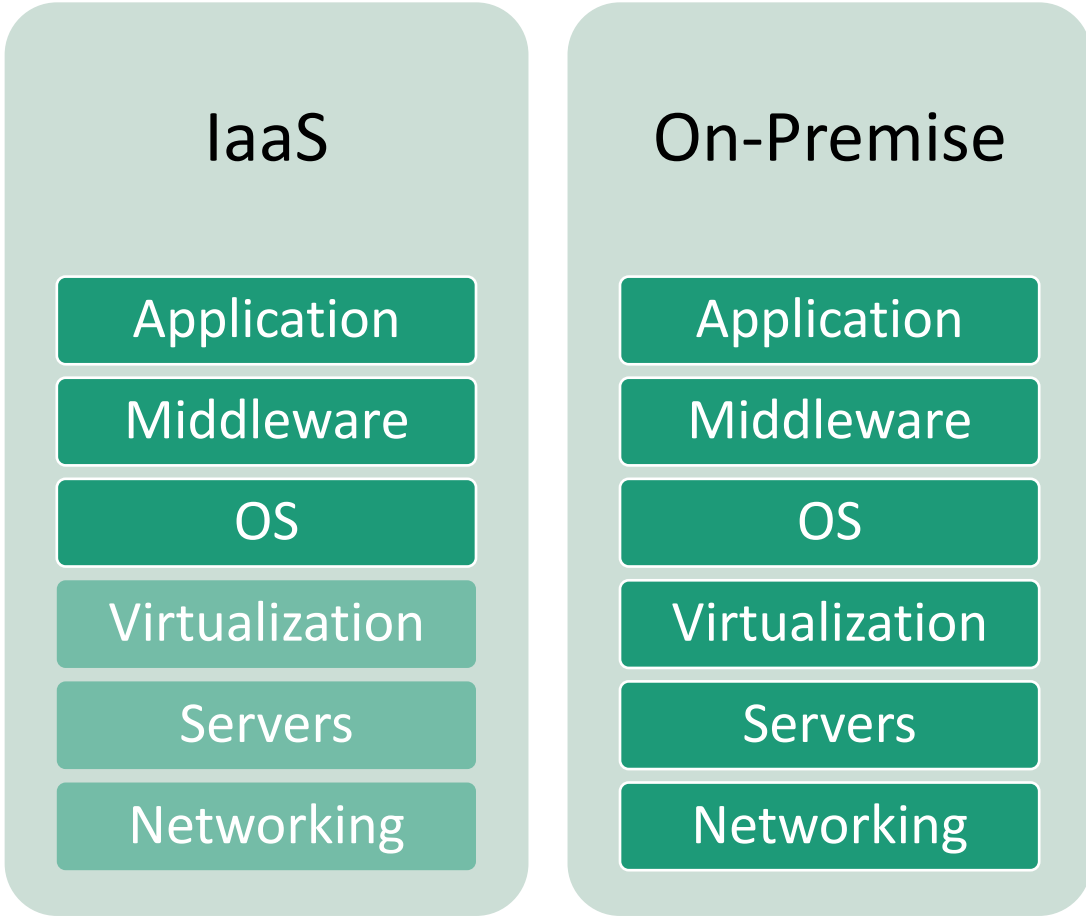


Chapter 2
Venturing in the **Cloud** Kingdom of **IaaS**

Exploring public cloud-based architectures

The IaaS approach

- You buy computing resources (virtual servers, storage, networking) from a Public Cloud Services provides, and use them to run your application.
- **On demand**
- With **pay-as-you-go** pricing



Key IaaS concepts

Service	Amazon Web Services	Microsoft Azure	Google Cloud
On-demand Virtual Servers	Elastic Compute Cloud (EC2)	Azure Virtual Machines	Compute Engine
Instance Scaling	Auto-scaling groups	Virtual Machine Scale Sets	Compute Engine Autoscaler
Load Balancing	Elastic Load Balancing	Azure Load Balancer	Cloud Load Balancing

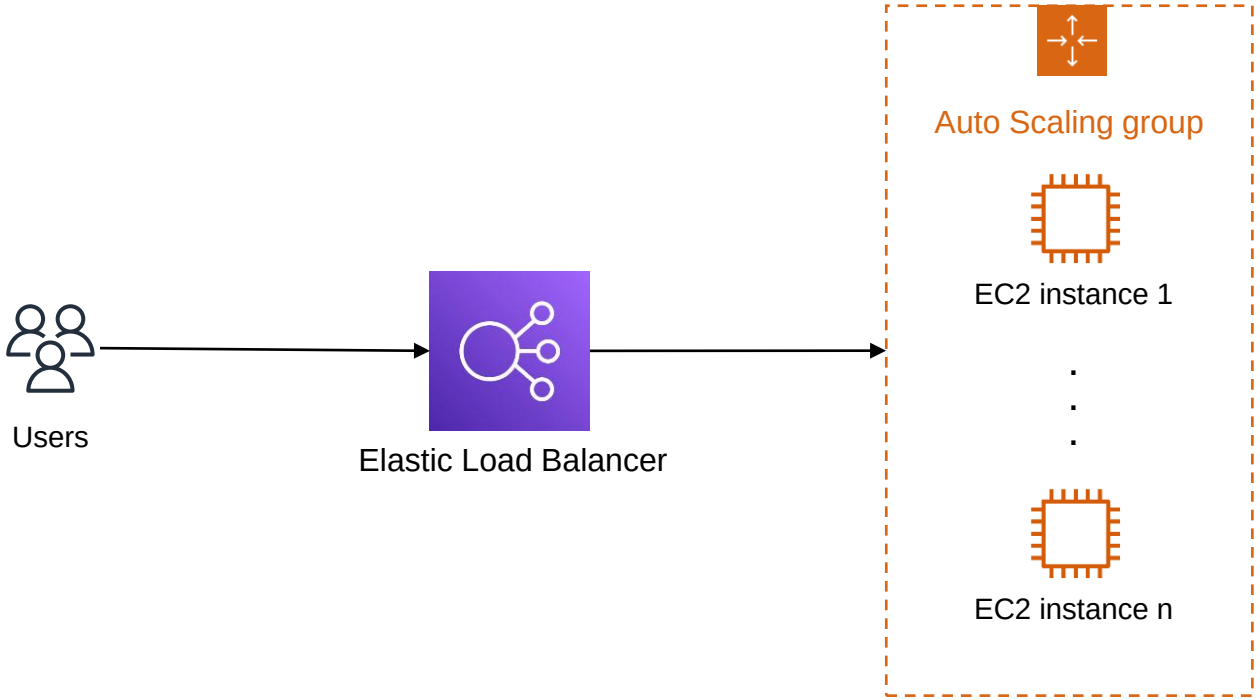
Azure for AWS Professionals: https://docs.microsoft.com/en-us/azure/architecture/aws-professional/
Google Cloud for AWS Professionals: https://cloud.google.com/docs/compare/aws

Instance Scaling in the Cloud

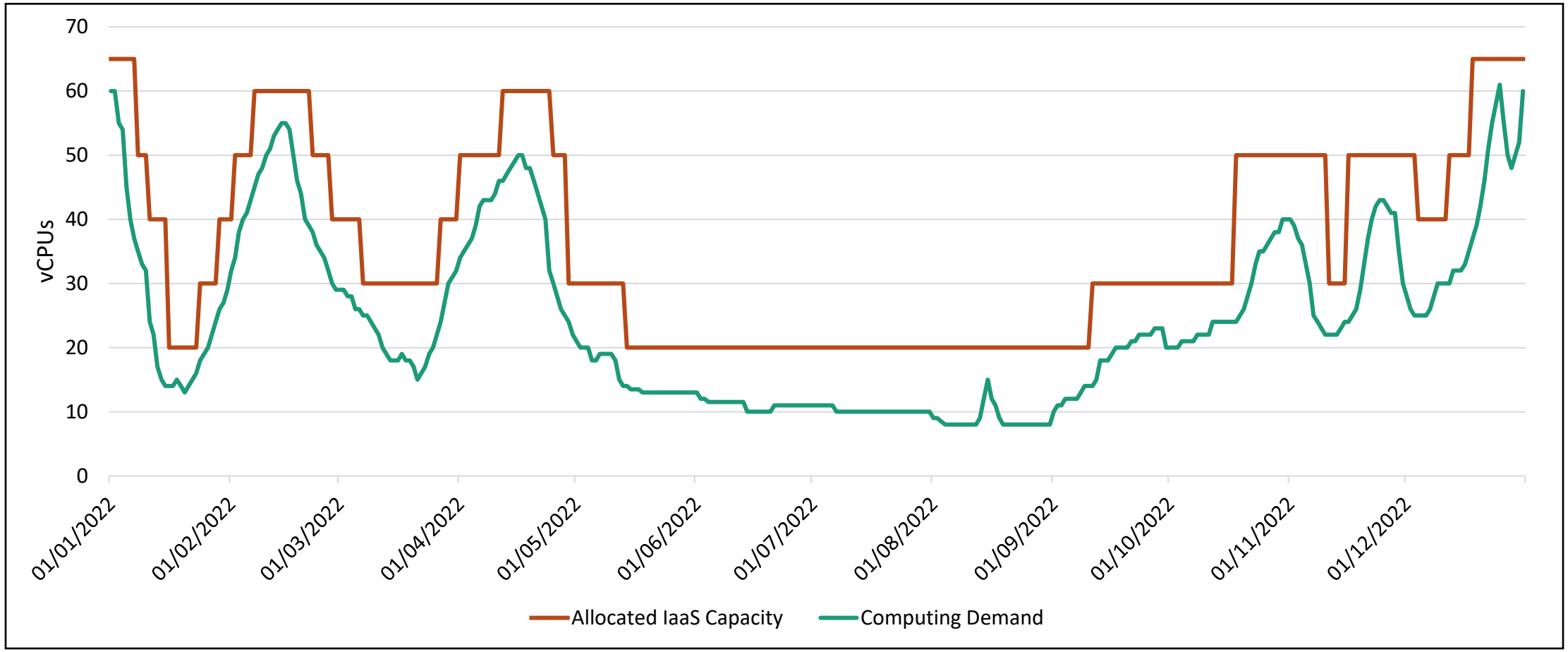
Typically one can set a minimum and a maximum number of instances, and three kinds of scaling plans are generally allowed:

- **Manual**, in which you manually require to scale up or to scale down your instances.
- **Scheduled**, in which you require to scale your instances at scheduled times.
- **Dynamic**, in which you specify a policy to regulate scaling. You can create policies based on dynamic properties, such as average CPU or memory utilization, number of served requests, etc…

Basic Elastic Architecture on AWS

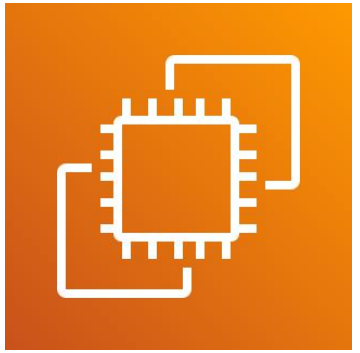


Handling unbalanced workloads in the Cloud



What if we need to deploy microservices?

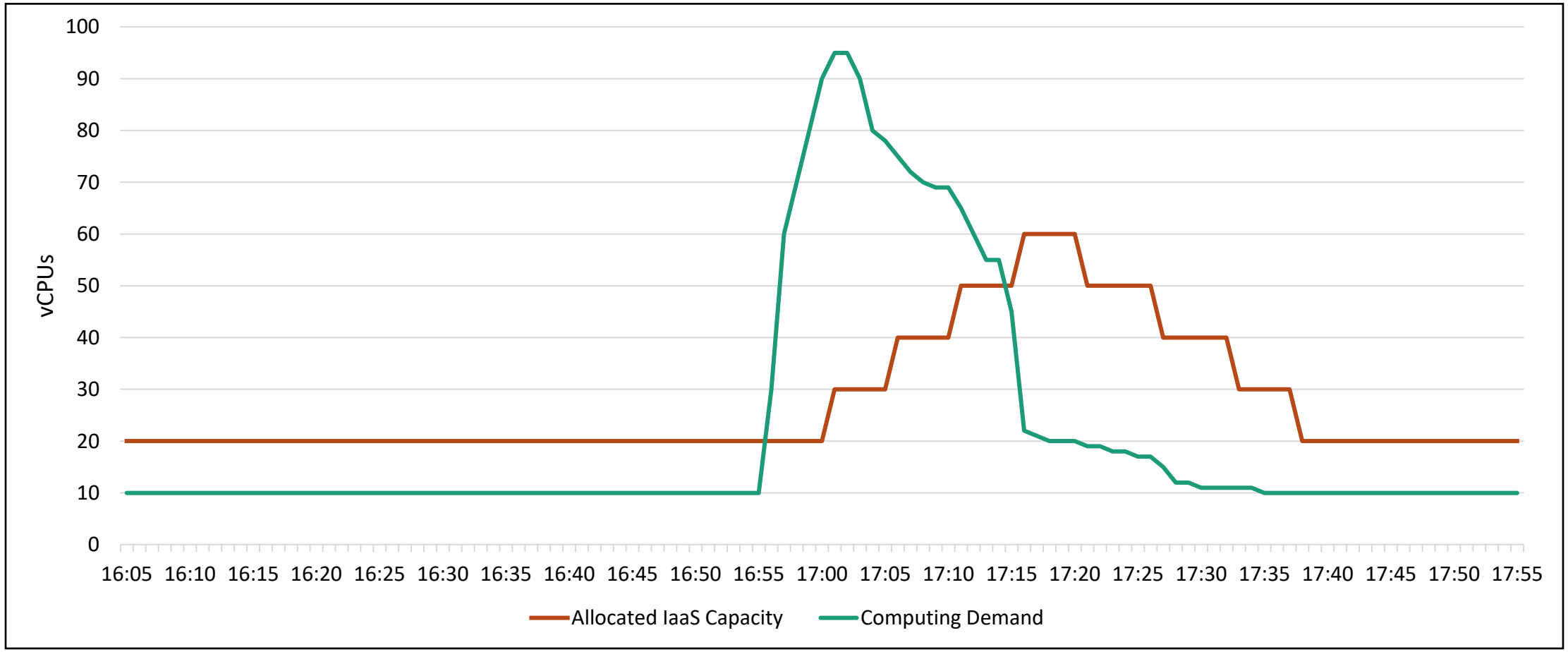
- Basically, that’s the same thing.
- You buy one or more Virtual Servers, set them up with your container engine, and run your containers.
- With AWS, you can use Elastic Container Service (ECS) to orchestrate your containers;
- ECS can run your containers in a customizable cluster of EC2 instances.



All seemed fine, until one day…

- It’s July 15th 2022. It’s a Friday. It’s 5 pm. Greetic is up and running with its 20 allocated vCPUs. Everything’s fine, and you’re about to head out for the weekend.
- Your phone rings. Many users are reporting connectivity issues and making fun of the service on Twitter.
- Your (totally not evil) boss is livid! What about the elasticity you promised with that Cloud stuff?

Handling unpredictable traffic spikes



Riddles for the adventurers

Are Cloud-based solutions **elastic**?

- Way more elastic than on-premise solutions, if you use the right services and configurations. But not elastic enough to handle unpredictable workload spikes!

Chapter 3
The ethereal **Serverless** realm

Understanding the novel serverless trend

What's all this **FaaS** about?

- **F**unctions **a**s **a** **S**ervice.
- FaaS services allow developers to run code without thinking about servers, provisioning, load balancing, scaling…
- You write the code, and say when it should run (event-driven).
- Code is run in **ephemeral** containers.
- Pay only for **actual** execution time.
- AWS Lambda, Google Cloud Functions, Azure Functions.

AWS Lambda

- Native support for Java, Go, C#, Python, Node.js, Ruby, Powershell.
- 128 to 10240 MB of memory.
- 500 MB of ephemeral disk space.
- Up to 15 minutes of computation.
- Up to hundreds of thousands of concurrent executions.
- $0,20 per 1M requests and $0.0000166667 for every GB-second (pricing)

Serverless Architectures

When working on-premises or with IaaS, we had to think about infrastructure.

- How many (virtual) servers?
- What's the best scaling policy?
- How many EC2 instances should there be in my ECS cluster?

With **Serverless** architectures there is no need to think about infrastructure at all. It's all managed by someone else!

How to go Serverless?

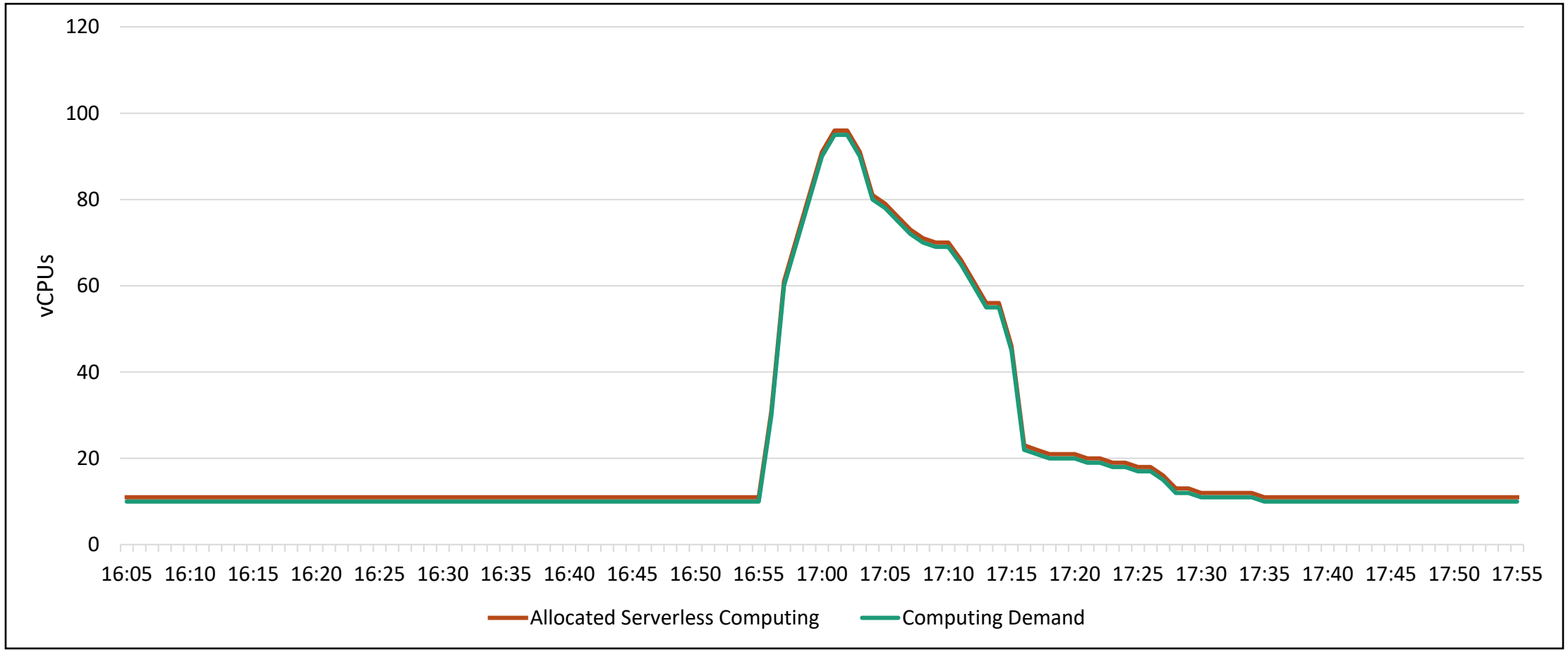
- Rely on third-party services so that traditional, always-on servers are not necessary.
  - (Mobile) Backend as a Service (MBaaS) services like Google Firebase, AWS Amplify, AWS Cognito, …
  - FaaS to run business logic on-demand.
- Typically associated with «smarter» clients.

Why Serverless?

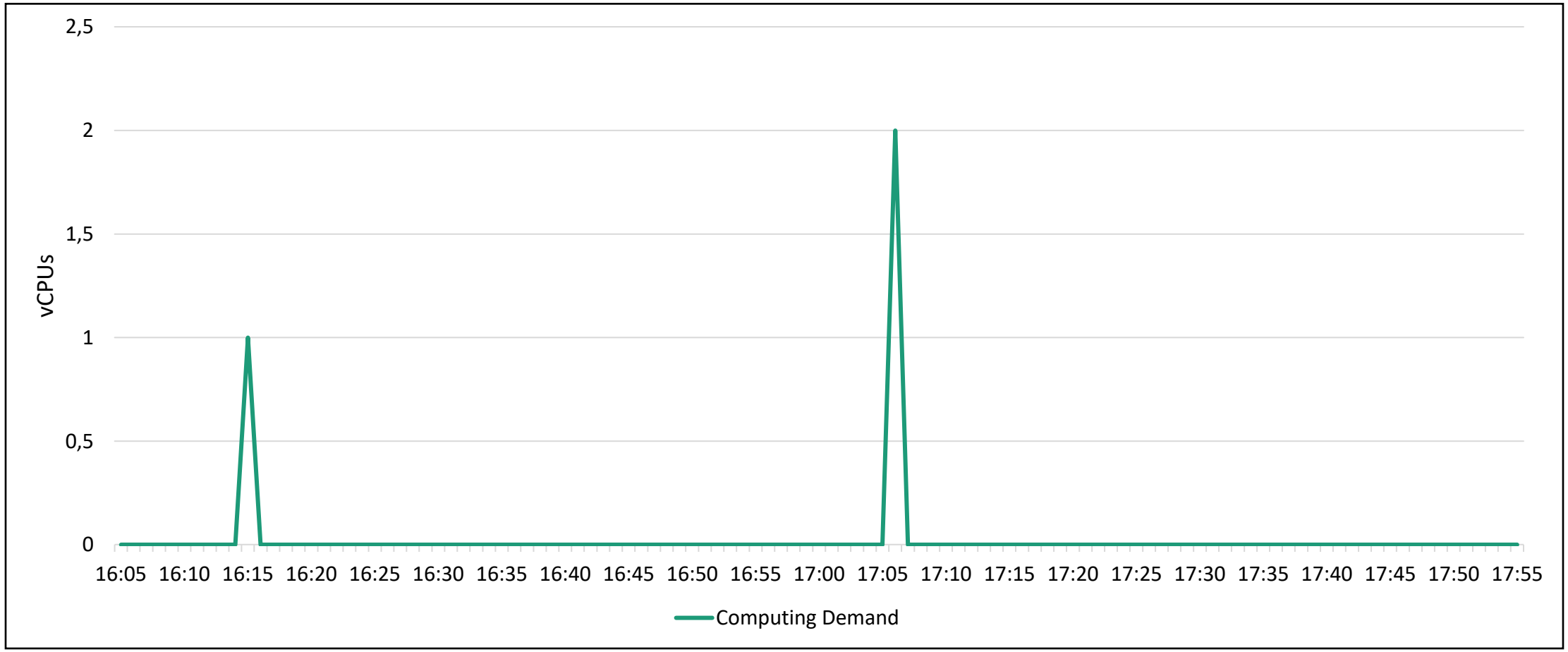
- ✓ No need to think about infrastructure: provisioning, scaling, load balancing;
- ✓ Less time to market.
- ✓ As elastic as it can get, at every time we use (and pay for) just as much computing resources as needed.
- ✓ Might be a lot cheaper: servers.lol

- ✗ Might not be applicable
- ✗ Might be more expensive
- ✗ Harder to test

Serverless use case: unpredictable spikes



Serverless use case: sporadic requests



Deploying a Serverless Microservice on AWS

ITSC2021 - mobility analysis fra | AWS Account | Workbench | AWS Management Console | AWS Fargate | Motore di calcol

https://console.aws.amazon.com/console/home?region=us-east-1#

Come iniziare | confs | dev | Learn | Starace, Luigi L.L. - Aut... | ITEE | Speeding up Mutation... | Alex-D/Cookies-EU-b... | RelevantMutantPredic... | ICSME20.pdf | Slack | Prime Music | PerformingRail | Altri segnalibri

aws | Services | Search for services, features, marketplace products, and docs [Alt+S] | vocstartsoft/user1358543=luigiliberolucio.starace@unina.it @ ... | N. Virginia | Support

AWS Management Console

AWS services

▼ Recently visited services

- DynamoDB
- Lambda
- EC2
- API Gateway
- Elastic Container Service
- AWS Amplify

▶ All services

Build a solution

Get started with simple wizards and automated workflows.

Launch a virtual machine
With EC2
2-3 minutes

Build a web app
With Elastic Beanstalk
6 minutes

Build using virtual servers
With Lightsail
1-2 minutes

Stay connected to your AWS resources on-the-go

AWS Console Mobile App now supports four additional regions. Download the AWS Console Mobile App to your iOS or Android mobile device. Learn more

Explore AWS

Build Serverless Apps with Infrastructure as Code

AWS Serverless Application Model (SAM) helps you build, debug, and deploy serverless apps. Learn more

Automate Document Data Processing

Extract text and understand the sentiment across millions of documents. Learn more

17:40
14/04/2021

Take Home Messages

- Scalability and Elasticity
- Pros and Cons of
  - On-prem
  - IaaS
  - Serverless Architectures

So, we're back at the beginning of our journey. What's the «right» architecture to use?

- ~~Ehm… The best architecture is the friends we made along the way?~~
- Well, that depends! But now we know more so we can make a concious choice ;)

References

- Martin Fowler. “*Who needs an architect?*” *IEEE SOFTWARE* 20.5 (2003): 11-13. https://martinfowler.com/ieeeSoftware/whoNeedsArchitect.pdf
- AWS. “*Serverless on AWS – Build and run applications without thinking about servers*” https://aws.amazon.com/serverless/
- AWS. “*Serverless Architecture with AWS Lambda – Overview and Best Practices*” https://d1.awsstatic.com/whitepapers/serverless-architectures-with-aws-lambda.pdf