

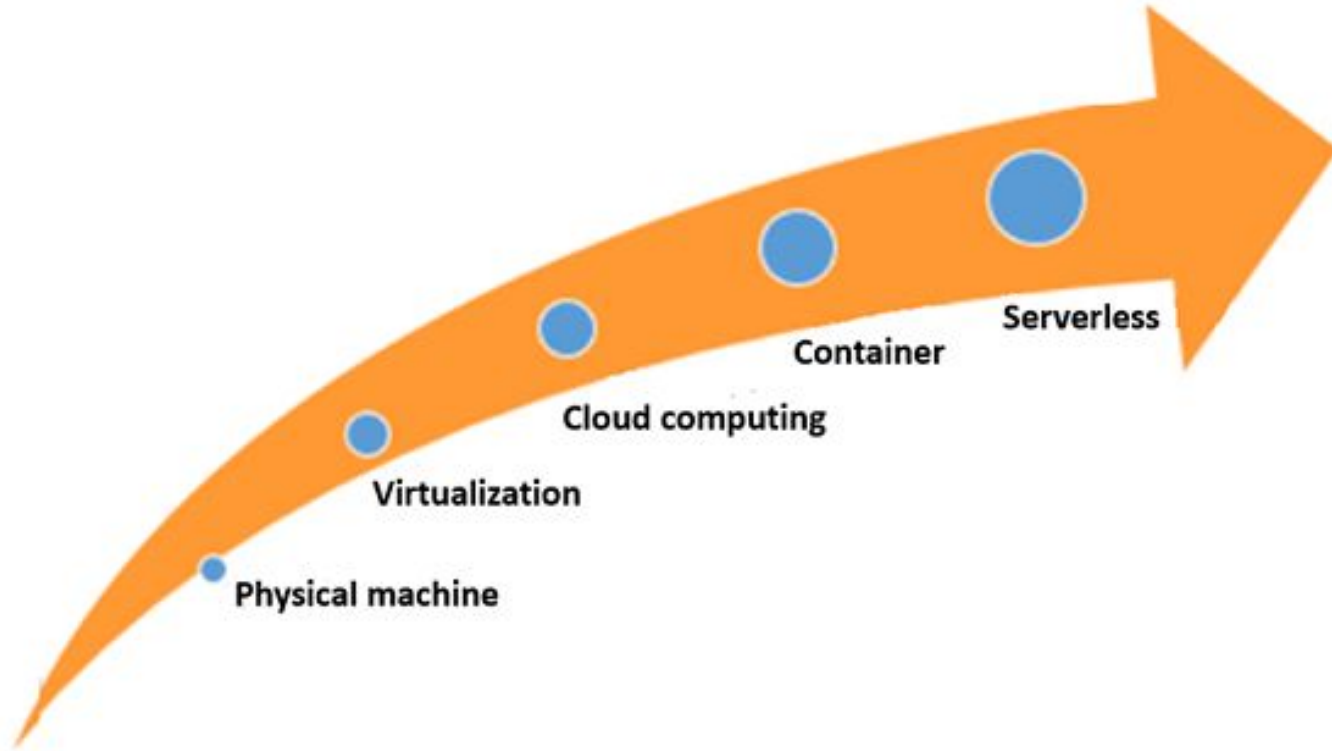
# Serverless Applications

...

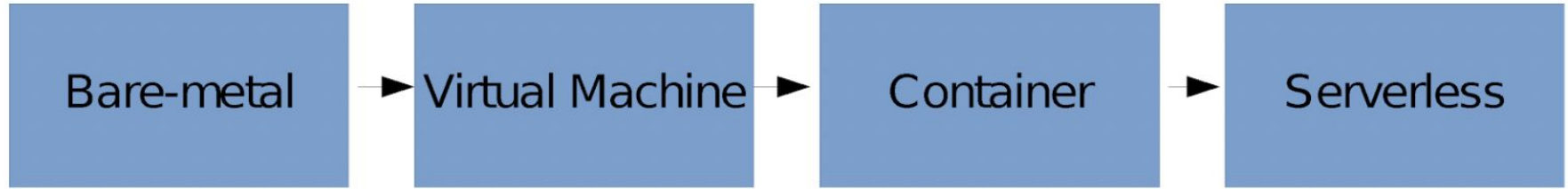
# What is serverless?

- Ability to deploy without thinking about servers
- Don't pay for instances
- Just pay per execution
  - How many times was my code run?

# History



# Levels of Abstraction



# Serverless

- Instead of deployments, you write functions
- Functions are the units – the lowest-level building block
- Typical workflow:
  - Write Code and unit tests
  - Upload Code as a “serverless function”
  - Run function
    - Triggered by API call or other event

# Pros of Serverless

- No servers to manage
- No thinking about patches and security
- Autoscaling seamless – and not developers' problems
- Pay per invocation
  - When no traffic, no charges
  - Simple pricing
- Service integrations
  - Very easy to integrate with other services within cloud provider
- Easy to debug and use

# Cons of Serverless

- Gain flexibility and ease of use **at the expense of control**
- Do not have visibility into infra
- Depending upon cloud provider to do things for you
  - Large companies will be partly serverless, partly not

# Cloud Providers: Serverless

- <https://cloud.google.com/serverless>
- <https://aws.amazon.com/lambda/>
- <https://azure.microsoft.com/en-us/solutions/serverless/>
- <https://www.digitalocean.com/blog/introducing-digitalocean-functions-serverless-computing>



# AWS Lambda Examples



Photograph is taken



**Amazon S3**  
Photo is uploaded  
to an S3 Bucket



Lambda is  
triggered



**AWS Lambda**  
Lambda runs image  
resizing code



Photo is resized into web,  
mobile, and tablet sizes

# AWS Lambda Examples



# AWS Lambda Examples



# Deploying Serverless

- Github with Lambda

<https://docs.timescale.com/tutorials/latest/aws-lambda/continuous-deployment/>

# In Class

- If you tried Django or Flask and were confused, serverless provides another option for deployment
- [Do this tutorial.](#)
- Two options:
  - Do the tutorial exactly as -is
  - Switch it to python and start adding your own code, as a start to the QuizApp
- Software engineering/architecture in practice is about choosing between various options and tools to do the same thing ;)