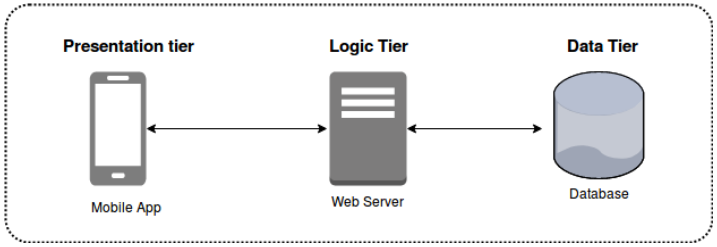
Serverless

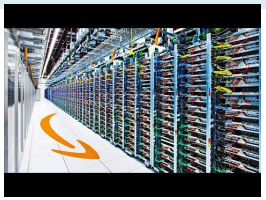
- Serverless computing
- API Gateway + lambda

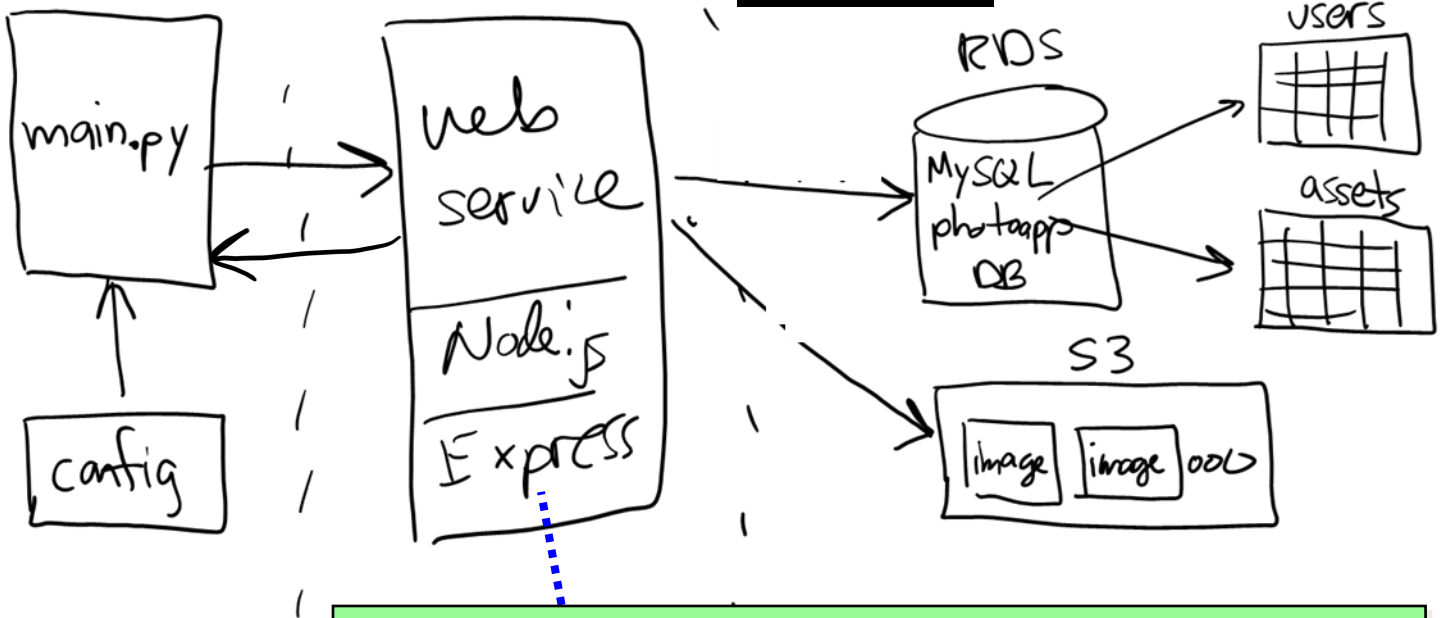
Monolithic multi-tier



- **Traditional software design for the cloud**
- **Monolithic approach --- one large code base on server**
  - *Safe, conservative engineering*
  - *No one gets fired for building systems this way :-)*

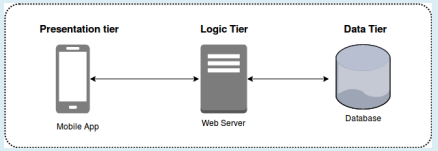
Project 02 --- monolithic web service






```
//
// PhotoApp web service
//
app.get('/stats', (req, res) => {…});
app.put('/user', (req, res) => {…});
app.get('/users', (req, res) => {…});
app.get('/assets', (req, res) => {…});
app.get('/bucket', (req, res) => {…});
app.get('/image/:assetid', (req, res) => {…});
app.post('/image/:userid', (req, res) => {…});
```

Alternative designs?



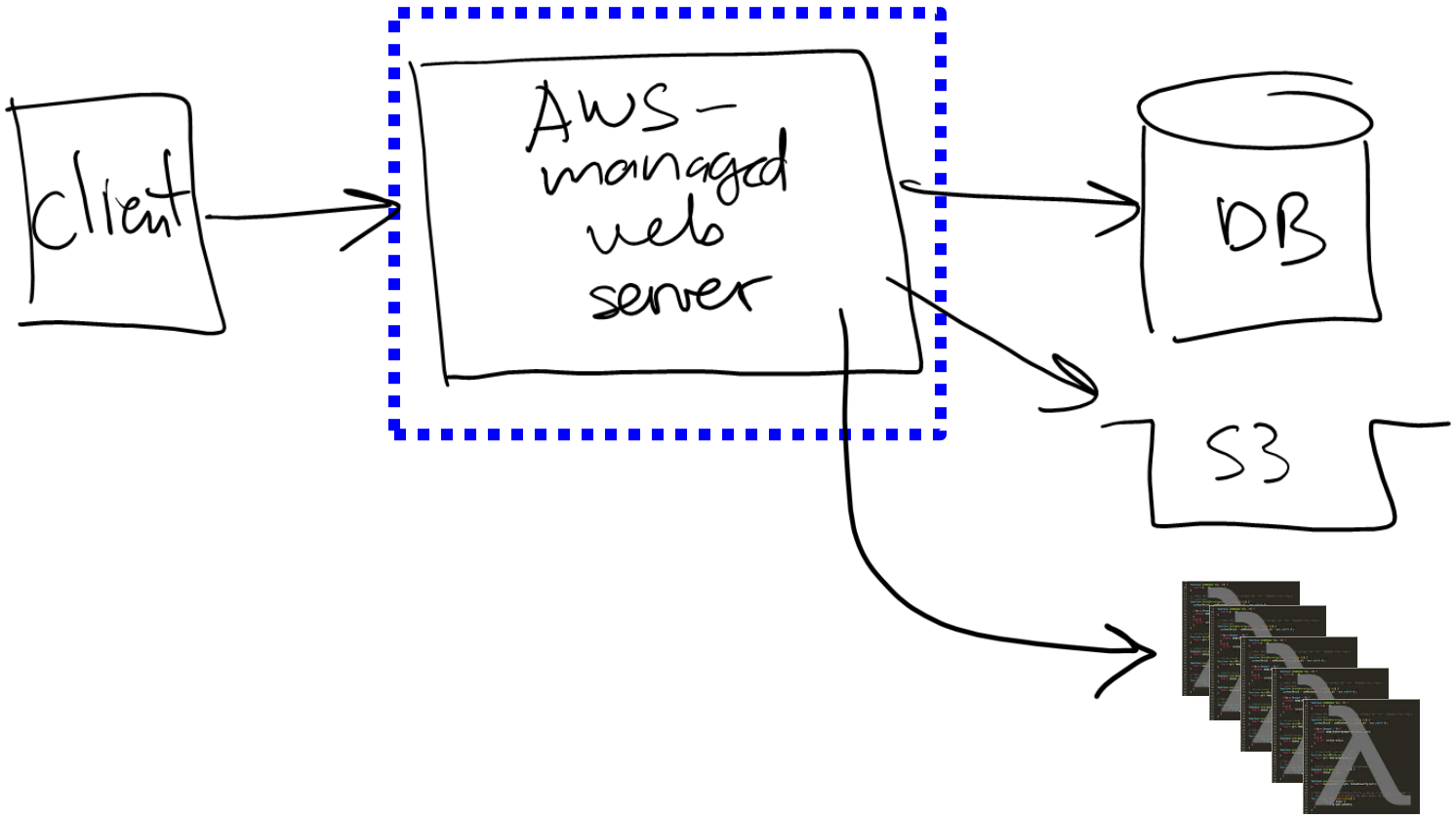
1. **Microservices**
   - *Break monolithic system apart --- easier to develop, update, release, but more moving parts to manage*
   - *Example: **Netflix** was one of the first to do this*

2. **Event-driven**
   - *Design based on events that occur / application states*
   - *Example: food delivery => menu, order, purchase, prepare, deliver*

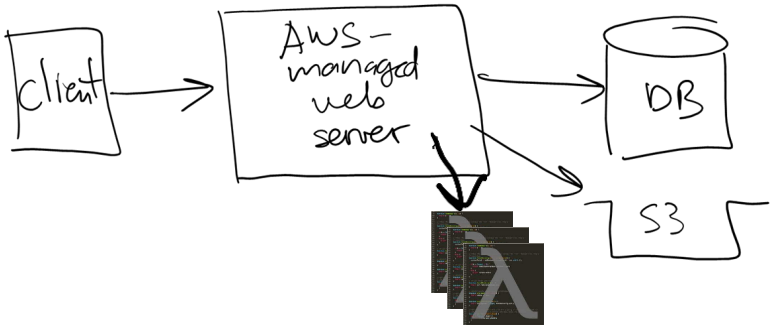
3. **Serverless computing…**

Serverless doesn't mean no server

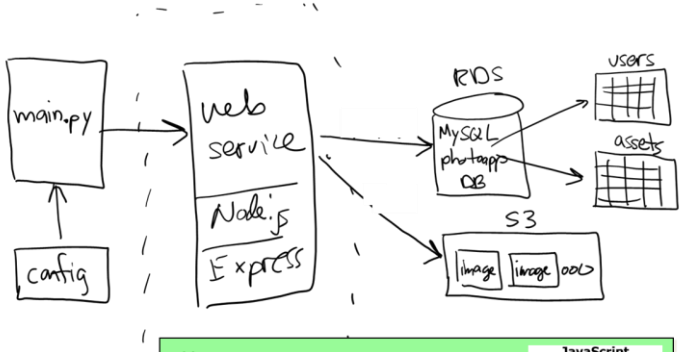
- **We still have a web server…**
- **We just don't manage it**



Why serverless?



- **Advantages:**
  1. Break monolithic code base into microservices / functions
     - *Enabling development in whatever language / platform makes the most sense --- JavaScript, Python, Java, …*
     - *Quicker to update and release functions / add new functionality*
  2. Retain advantages of web server tier but let AWS manage
  3. Scalability of server & functions w/o idle capacity (saving $)

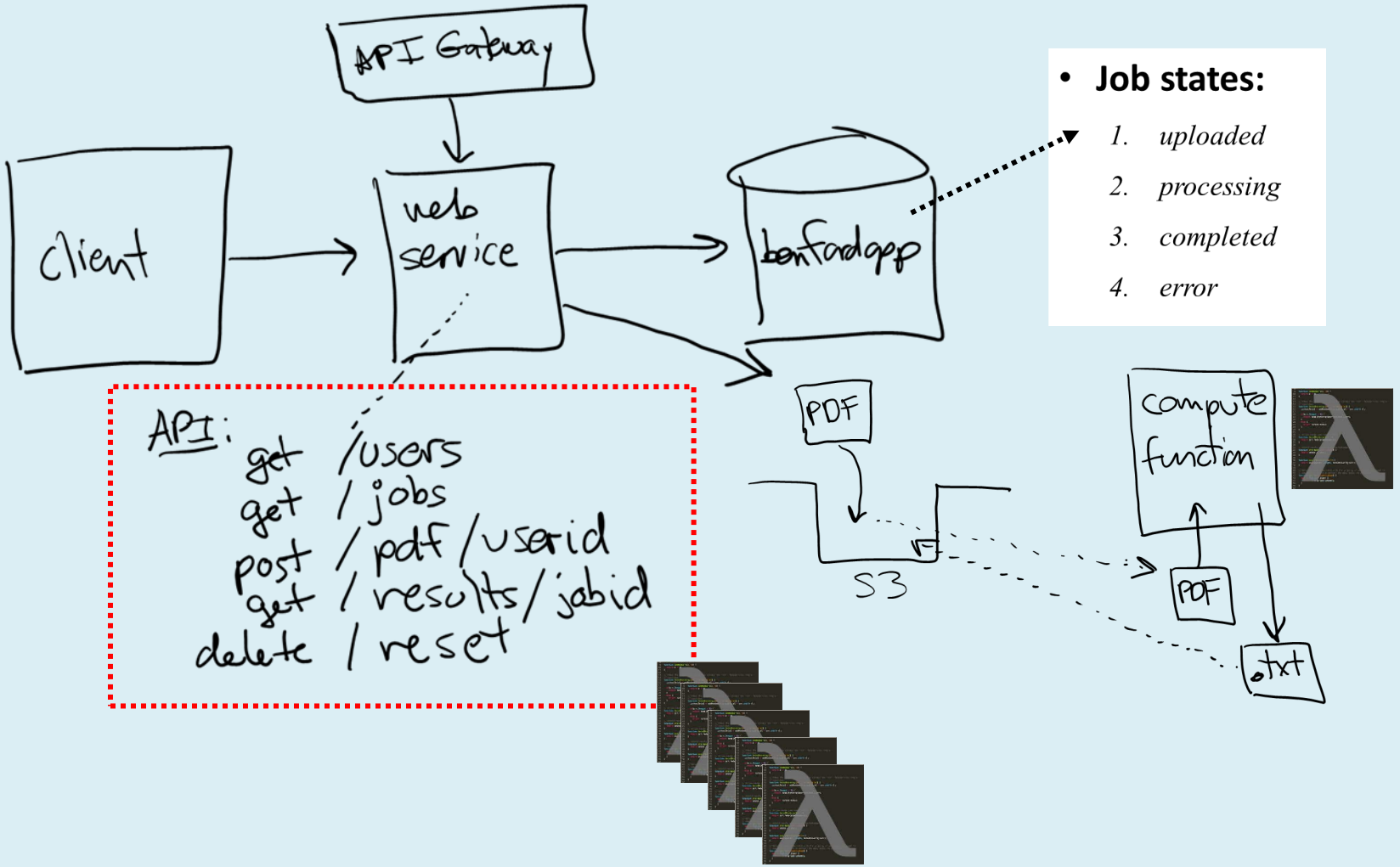



```
//
// PhotoApp web service
//
app.get('/stats', (req, res) => {…});
app.put('/user', (req, res) => {…});
app.get('/users', (req, res) => {…});
app.get('/assets', (req, res) => {…});
app.get('/bucket', (req, res) => {…});
app.get('/image/:assetid', (req, res) => {…});
app.post('/image/:userid', (req, res) => {…});
```

longer latency ☹

Example: Project 03

- *Serverless and event-driven…*



- **Job states:**
  1. *uploaded*
  2. *processing*
  3. *completed*
  4. *error*

AWS lambda

- **By far the simplest, least expensive way to compute**

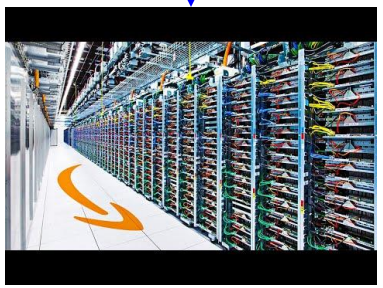
EC2, EKS, ECS, Fargate

Elastic Beanstalk

API Gateway + Lambda

Lambda

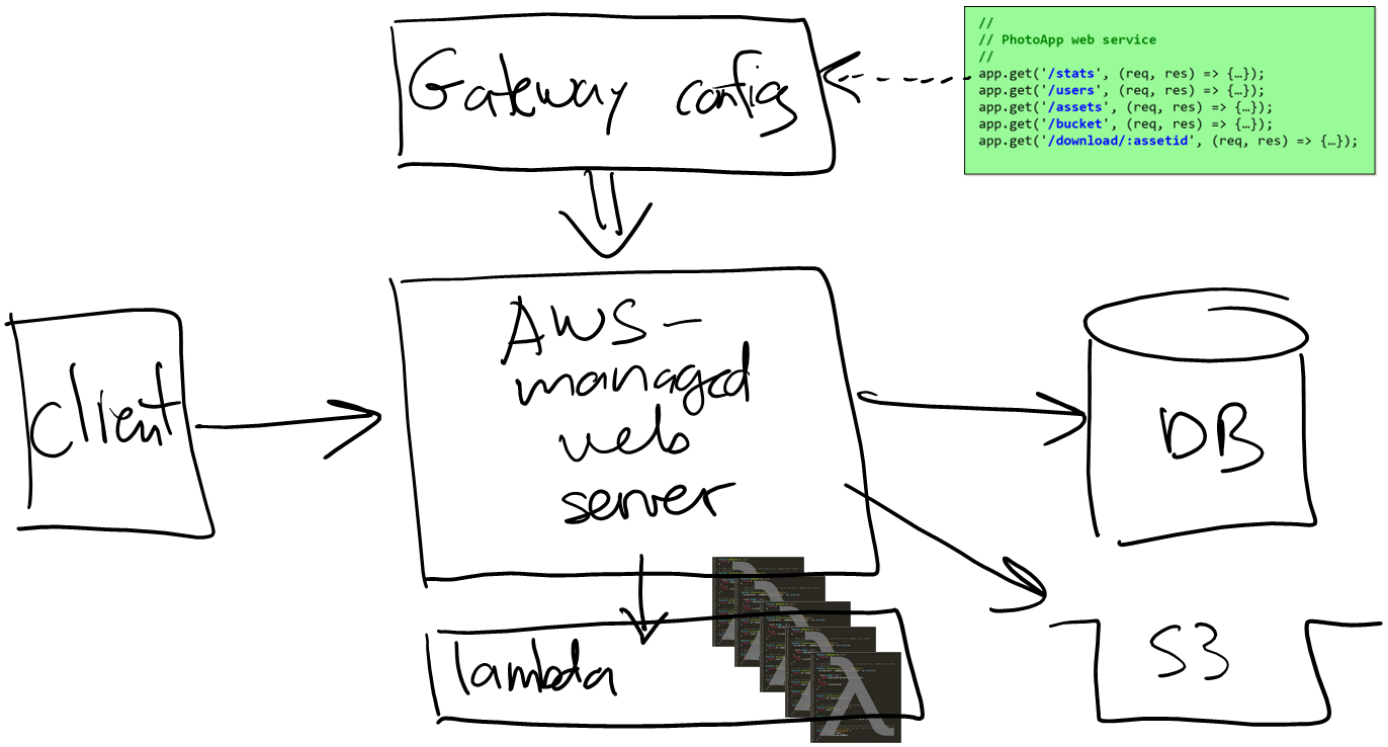
function



Serverless computing with API Gateway

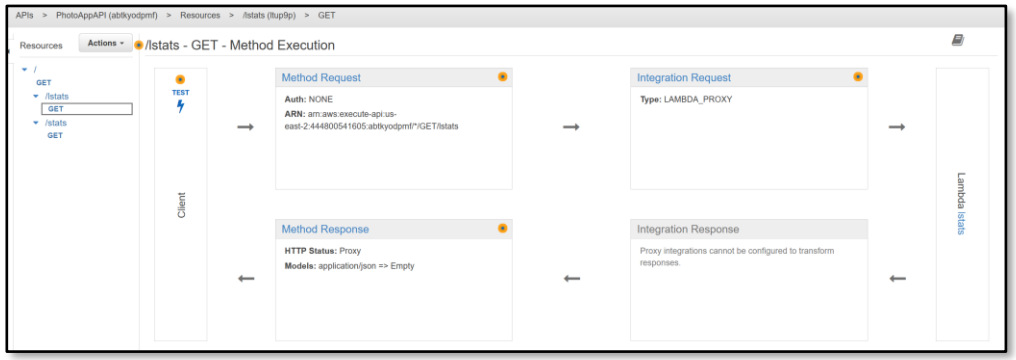
- **Just another step in the evolution of making AWS easier:**

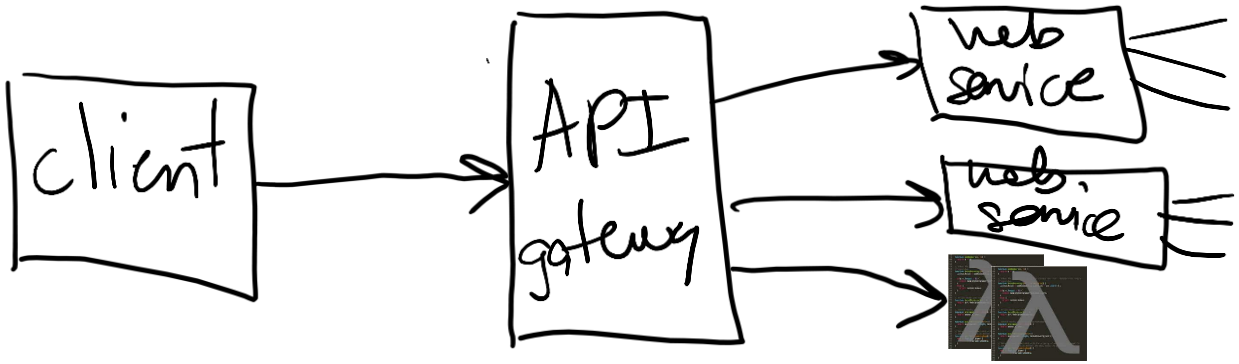
EC2, EKS, ECS, Fargate Elastic Beanstalk **API Gateway + Lambda**



API Gateway

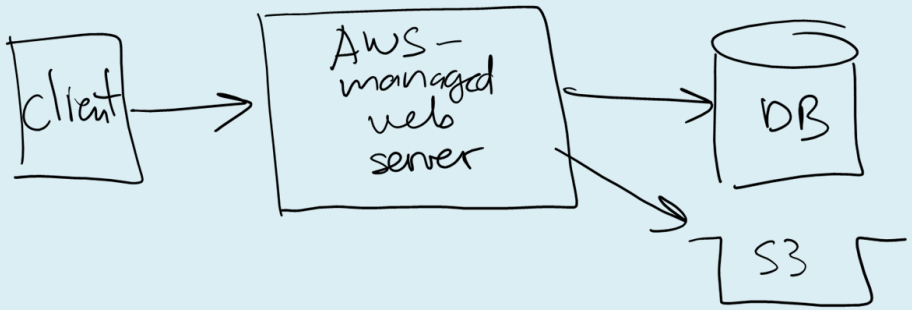
- **API Gateway allows you to define a RESTful API that forwards to other services / lambdas**
  - *Define HTTP verb and URL path (e.g. GET /movies)*
  - *Specify target…*





Need faster response (lower latency)?

- ***Replace lambda with faster technology (more $)…***



EC2, EKS, ECS, Fargate — Elastic Beanstalk — … — Lambda

That's it, thank you!