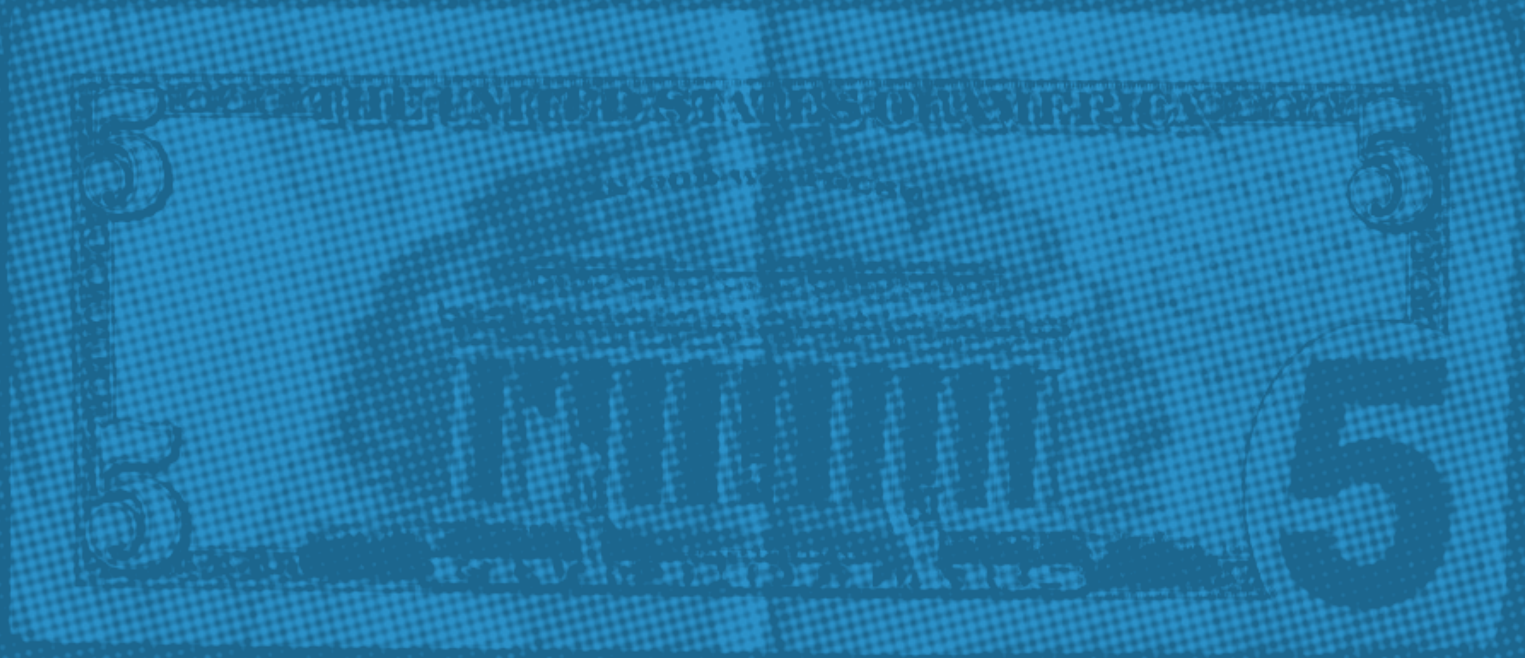
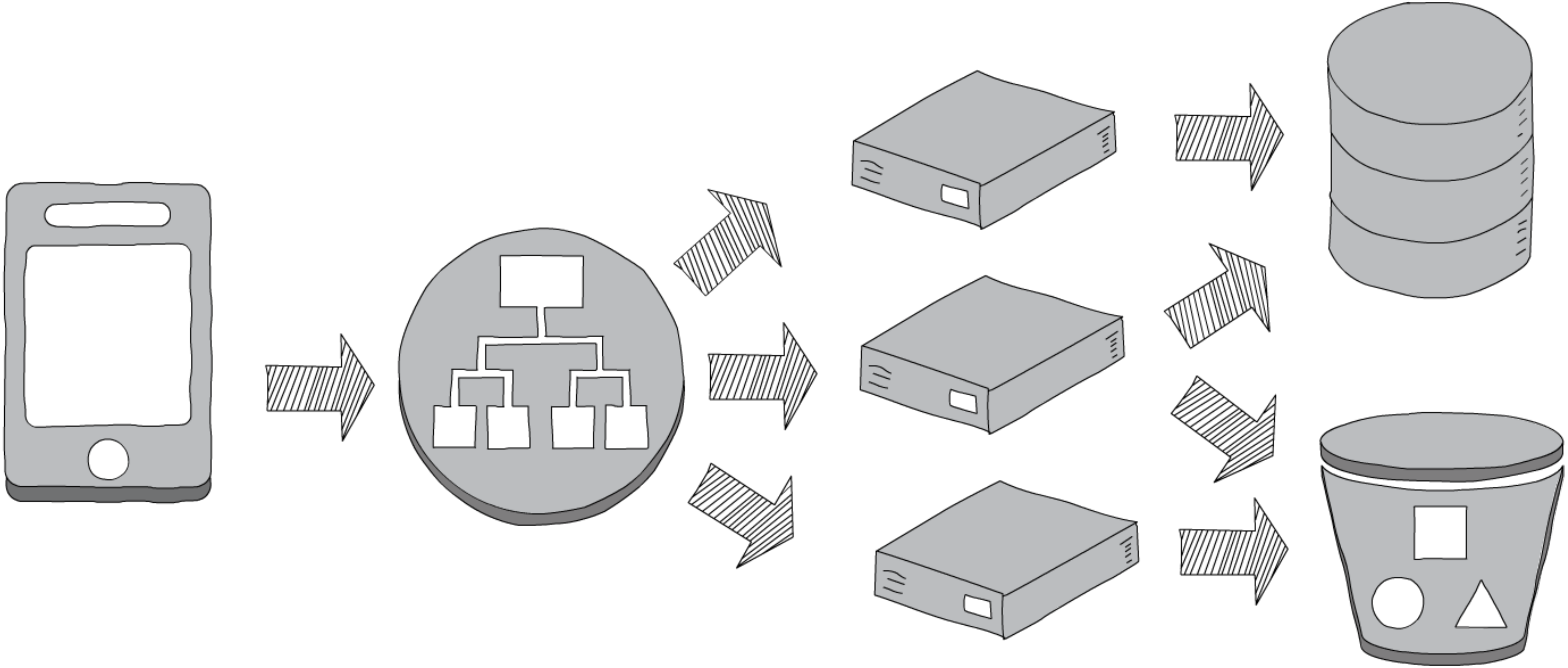
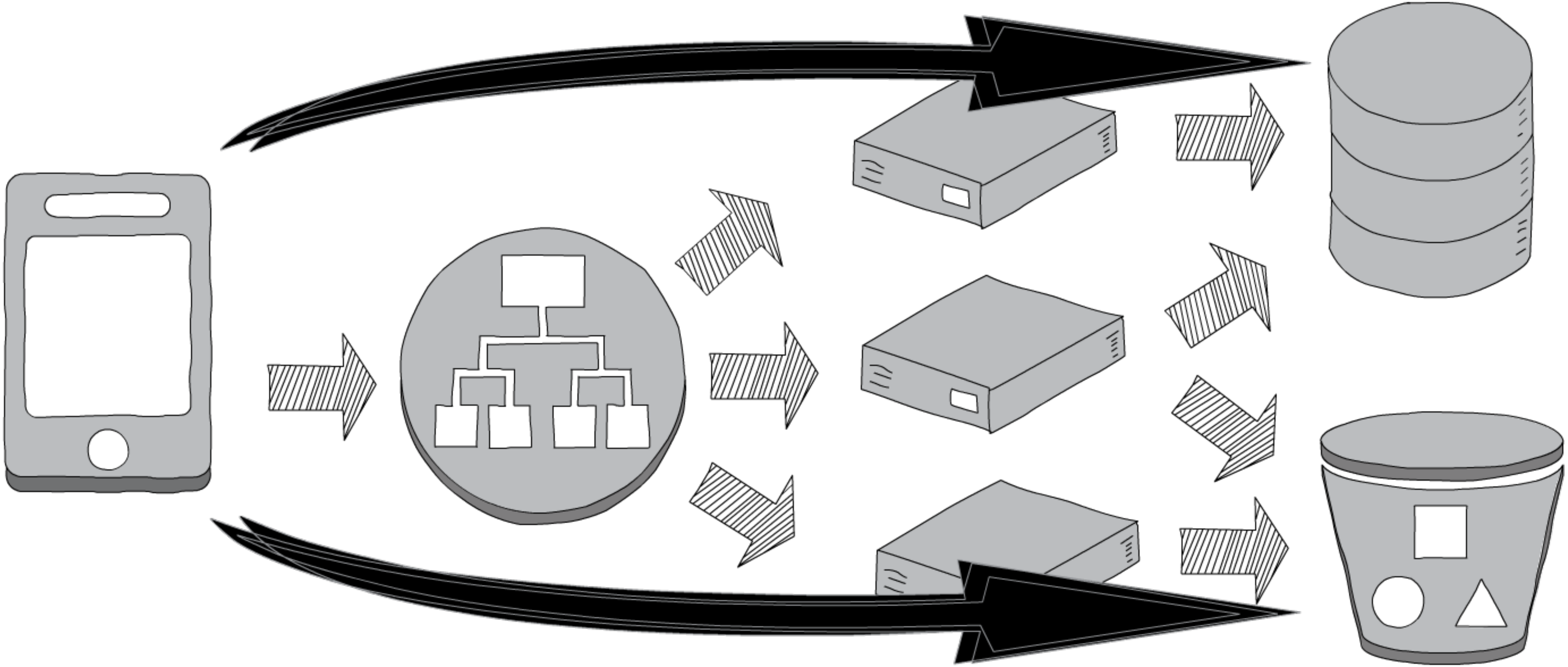
Five things you need to know about serverless



Gojko Adzic | https://gojko.net | @gojkoadzic





~~Serverless~~ Socketless

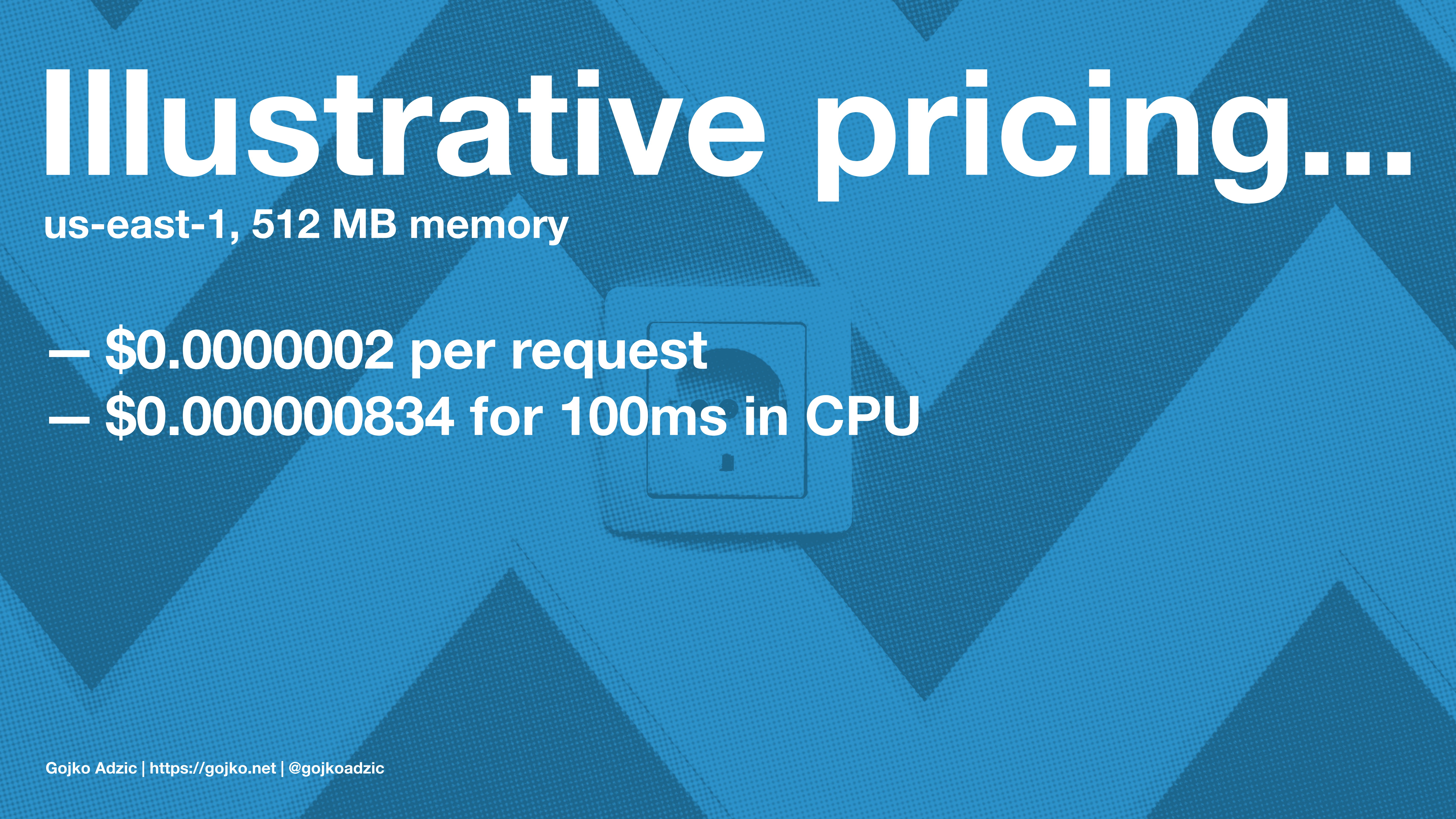
- Pricing
- Versions
- Task routing
- Application state
- Security

Paying for utilisation

not capacity

not environments

not service instances



Illustrative pricing...

us-east-1, 512 MB memory

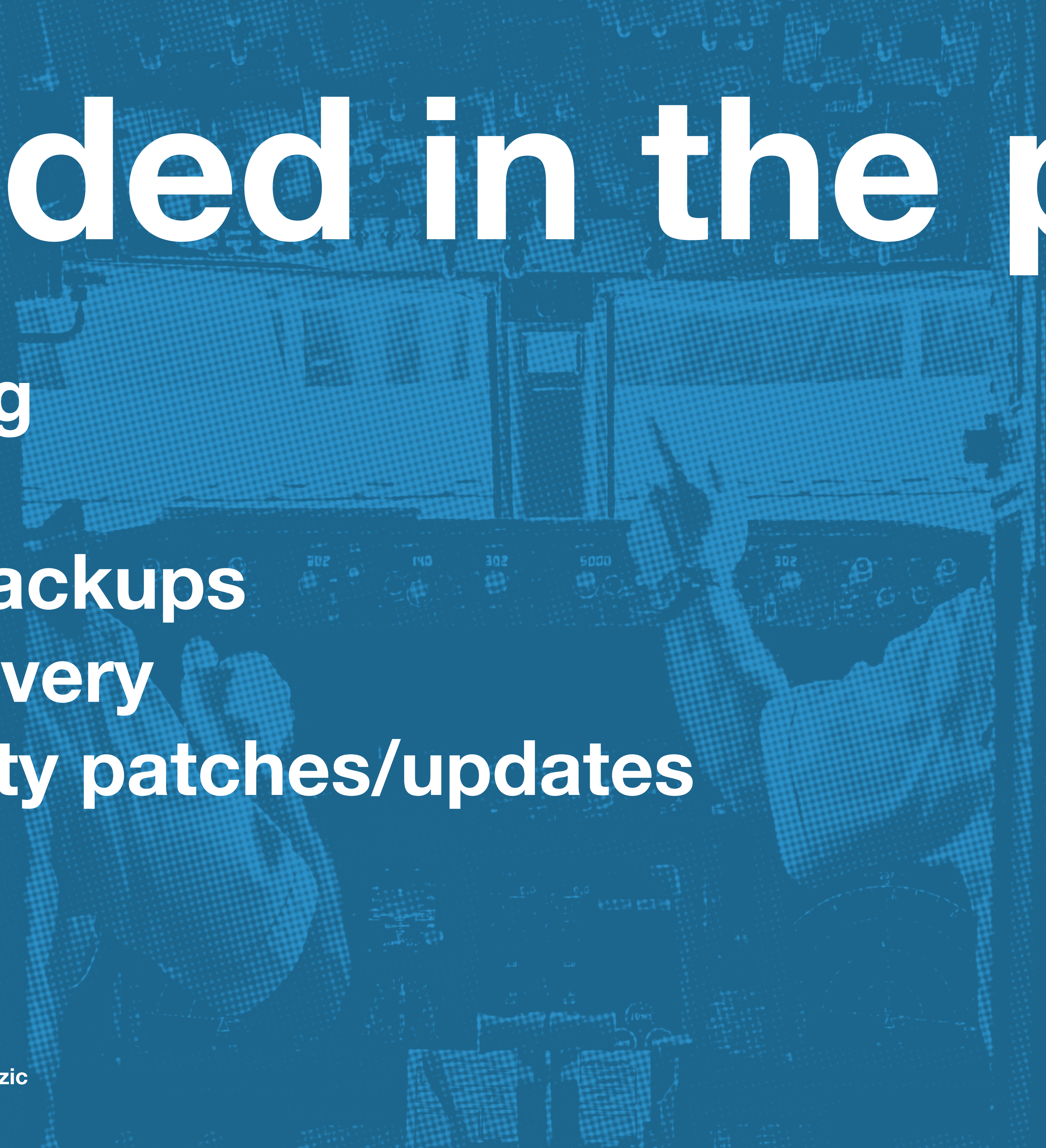
- $0.0000002 per request
- $0.000000834 for 100ms in CPU

monthly pricing...

- 100ms every 5 minutes = 1¢
- non-stop = $27
- EC2 primary + failover = $9

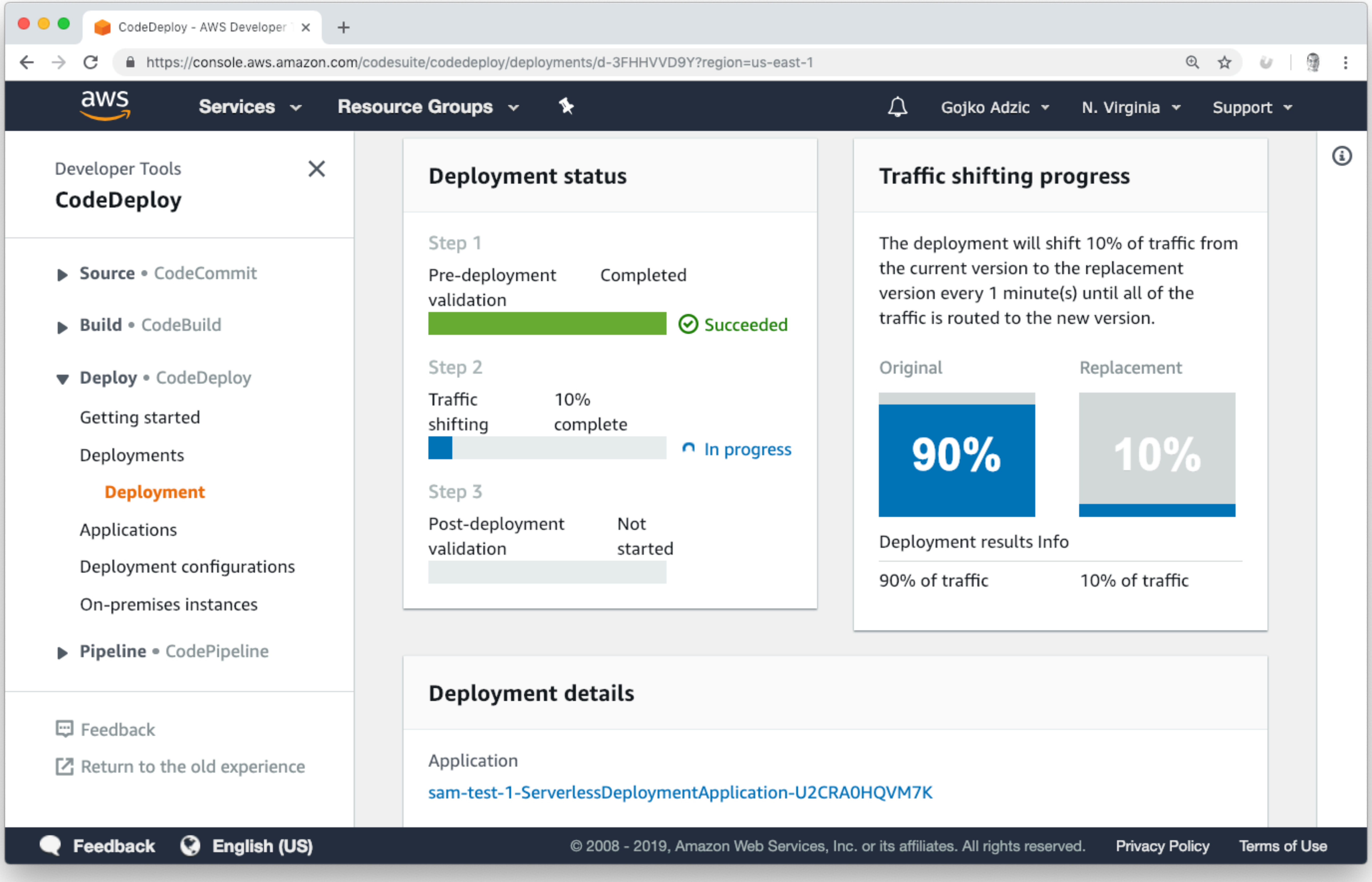
Included in the price

- monitoring
- scaling
- failover/backups
- error recovery
- OS security patches/updates



```
ConvertFileFunction:
  Type: AWS::Serverless::Function
  Properties:
    Timeout: 600
    DeadLetterQueue:
      Type: SNS
      TargetArn: !Ref NotifyAdmins
```

```
ConvertFileFunction:
  Type: AWS::Serverless::Function
  Properties:
    DeploymentPreference:
     Type: Canary10Percent10Minutes
     Alarms:
       - !Ref CheckForLambdaErrors
       - !Ref CheckForDropInSales
       - !Ref CheckForDropInConversion
     Hooks:
       PreTraffic: !Ref ClearStatisticsLambda
       PostTraffic: !Ref NotifyAdminsLambda
```



Deployment status

Step 1

Pre-deployment validation — Completed — Succeeded

Step 2

Traffic shifting — 10% complete — In progress

Step 3

Post-deployment validation — Not started

Traffic shifting progress

The deployment will shift 10% of traffic from the current version to the replacement version every 1 minute(s) until all of the traffic is routed to the new version.

Original	Replacement
90%	10%

Deployment results Info

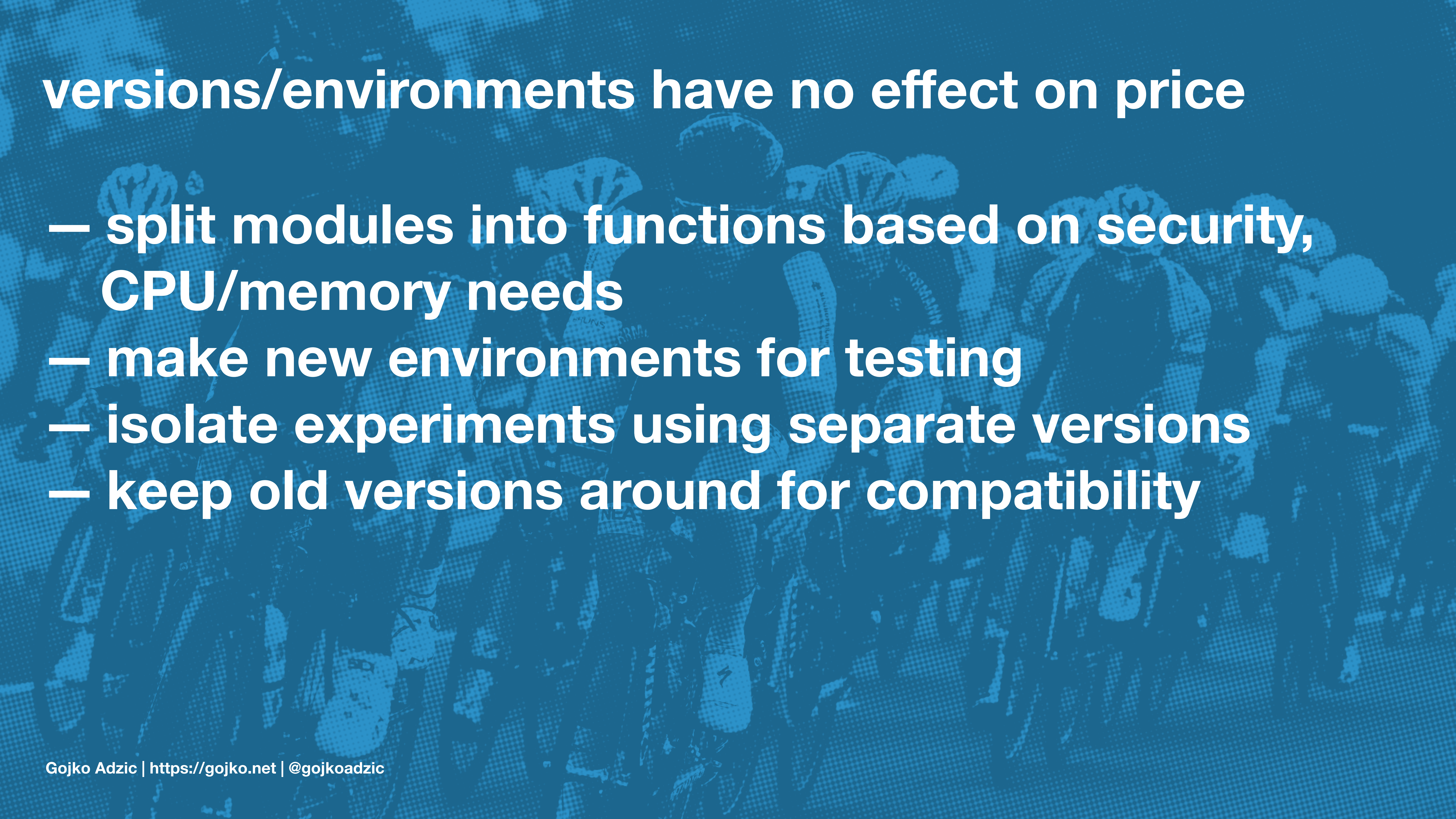
90% of traffic	10% of traffic

Deployment details

Application

sam-test-1-ServerlessDeploymentApplication-U2CRA0HQVM7K

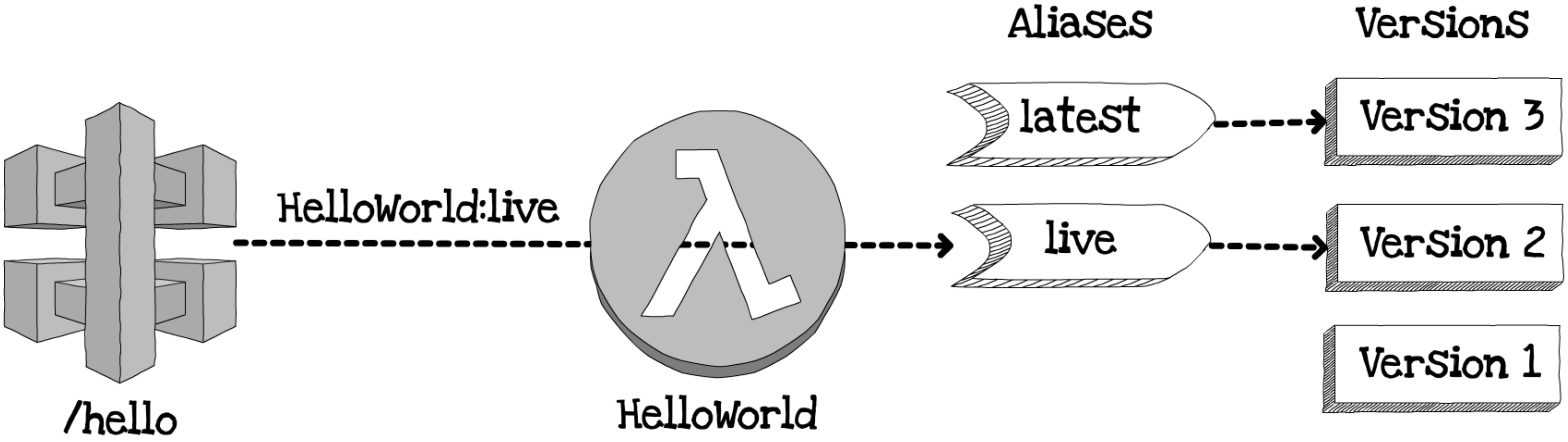
Versions/environments have no effect on price



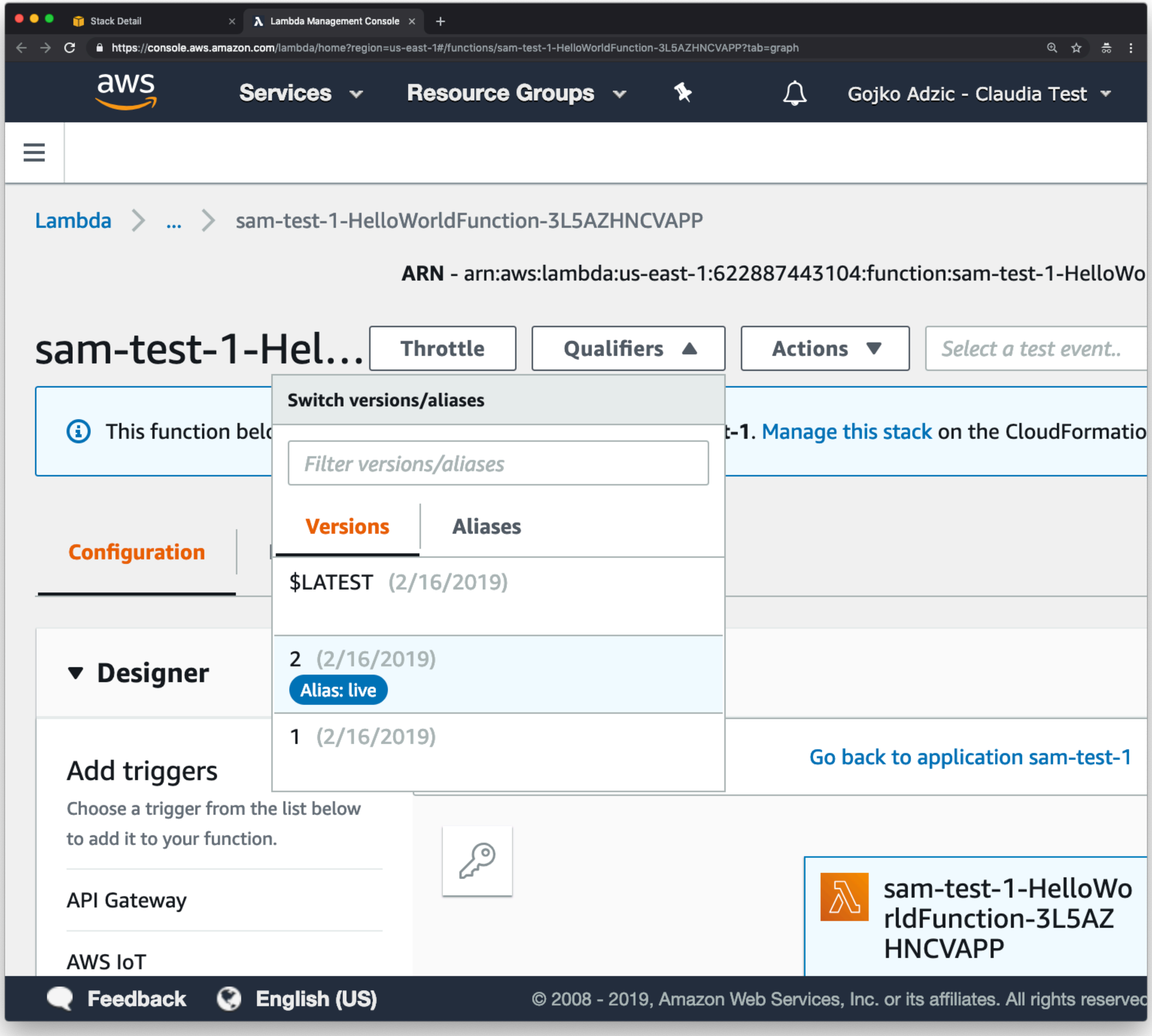
versions/environments have no effect on price

- split modules into functions based on security, CPU/memory needs
- make new environments for testing
- isolate experiments using separate versions
- keep old versions around for compatibility

Versions and aliases



```
ConvertFileFunction:
  Type: AWS::Serverless::Function
  Properties:
    AutoPublishAlias: live
```


Stack Detail
Lambda Management Console
https://console.aws.amazon.com/lambda/home?region=us-east-1#/functions/sam-test-1-HelloWorldFunction-3L5AZHNCVAPP?tab=graph
aws
Services
Resource Groups
Gojko Adzic - Claudia Test
Lambda
...
sam-test-1-HelloWorldFunction-3L5AZHNCVAPP
sam-test-1-Hel...
Throttle
Qualifiers
Actions
Select a test event..
Switch versions/aliases
Filter versions/aliases
Versions
Aliases
Configuration
Designer
Add triggers
API Gateway
AWS IoT
Go back to application sam-test-1
sam-test-1-HelloWorldFunction-3L5AZHNCVAPP
Feedback
English (US)

ARN - arn:aws:lambda:us-east-1:622887443104:function:sam-test-1-HelloWo

This function belo t-1. Manage this stack on the CloudFormatio

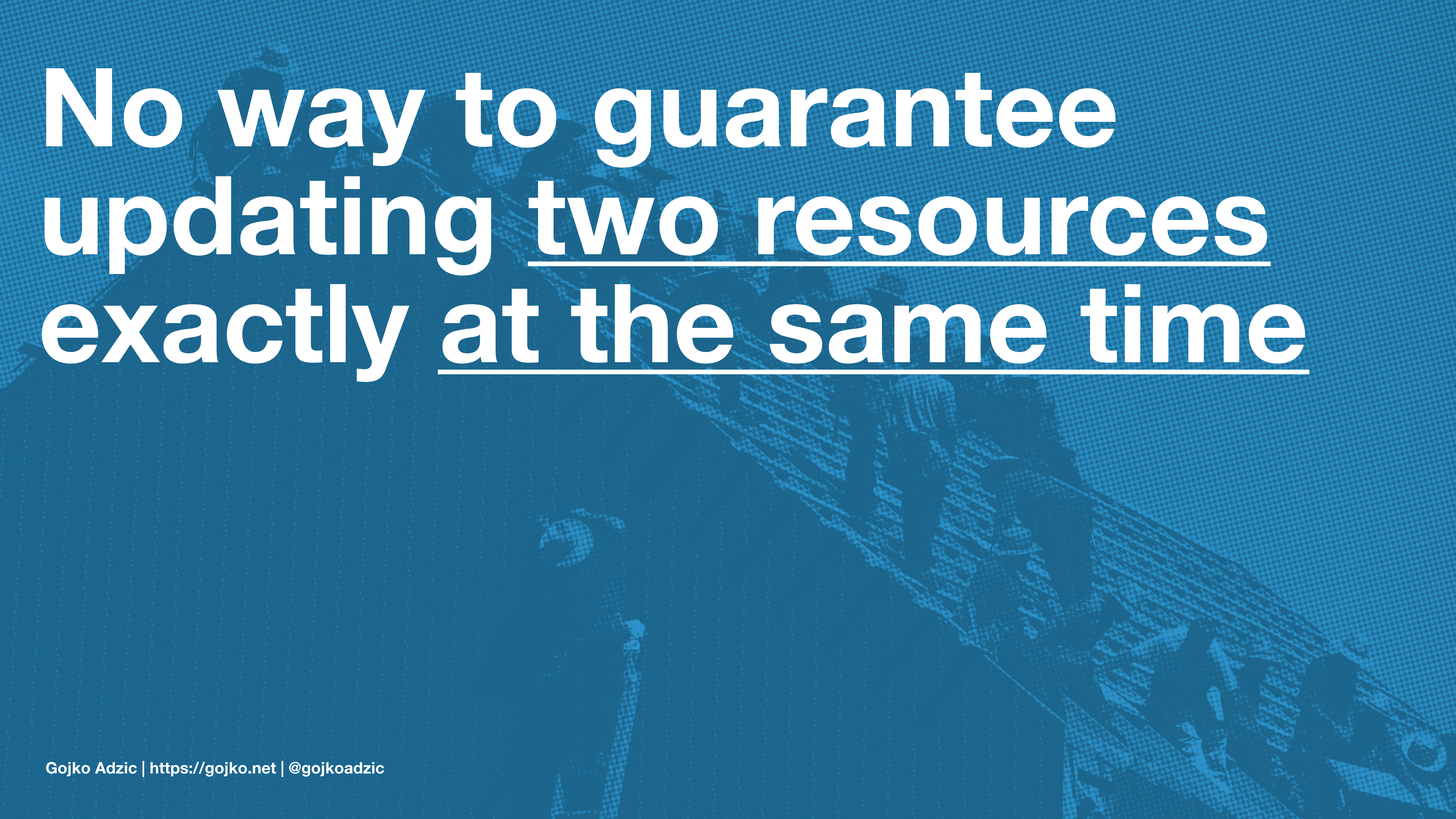
$LATEST (2/16/2019)

2 (2/16/2019)
Alias: live

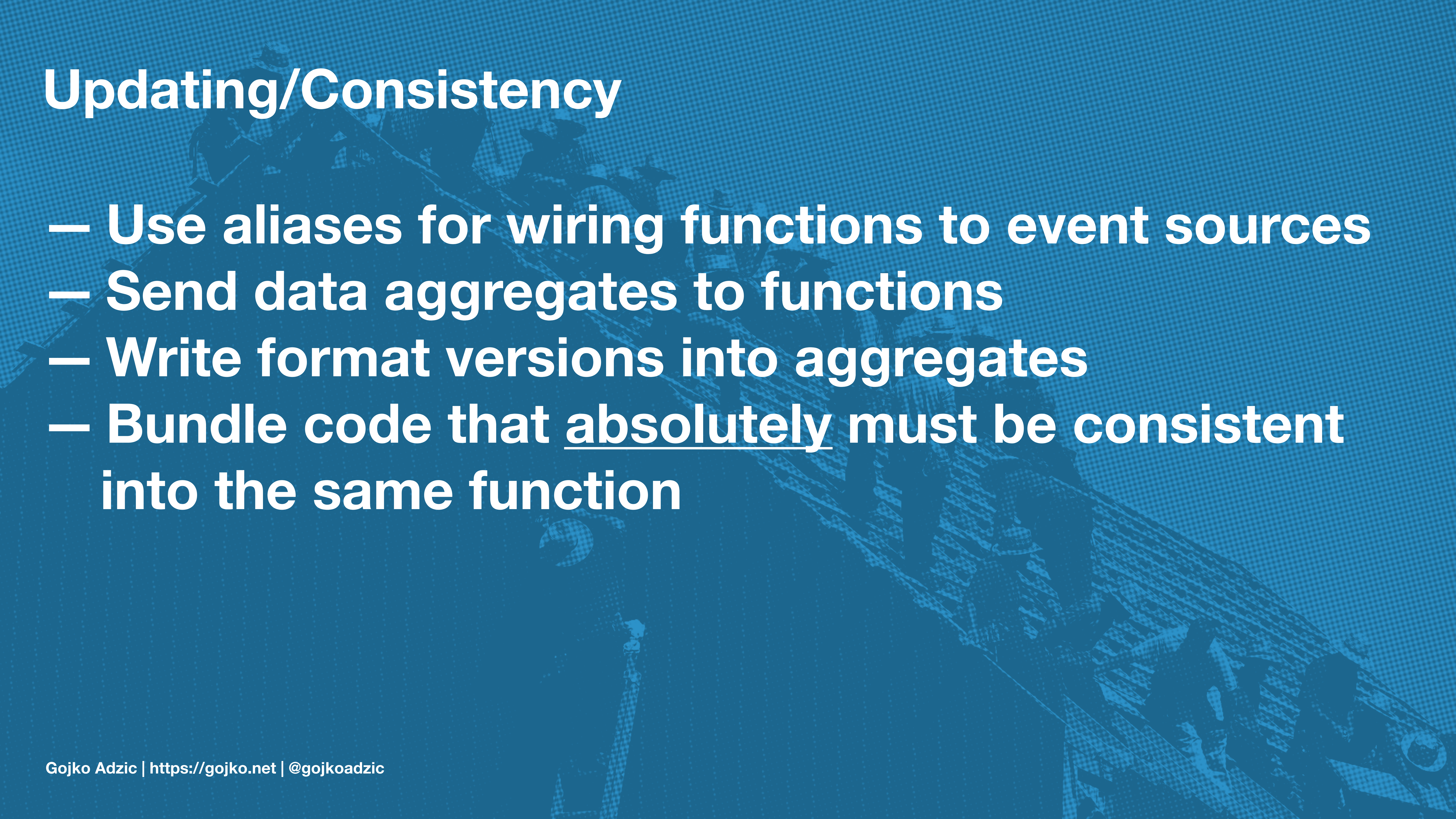
1 (2/16/2019)

Choose a trigger from the list below to add it to your function.

© 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved



No way to guarantee updating two resources exactly at the same time



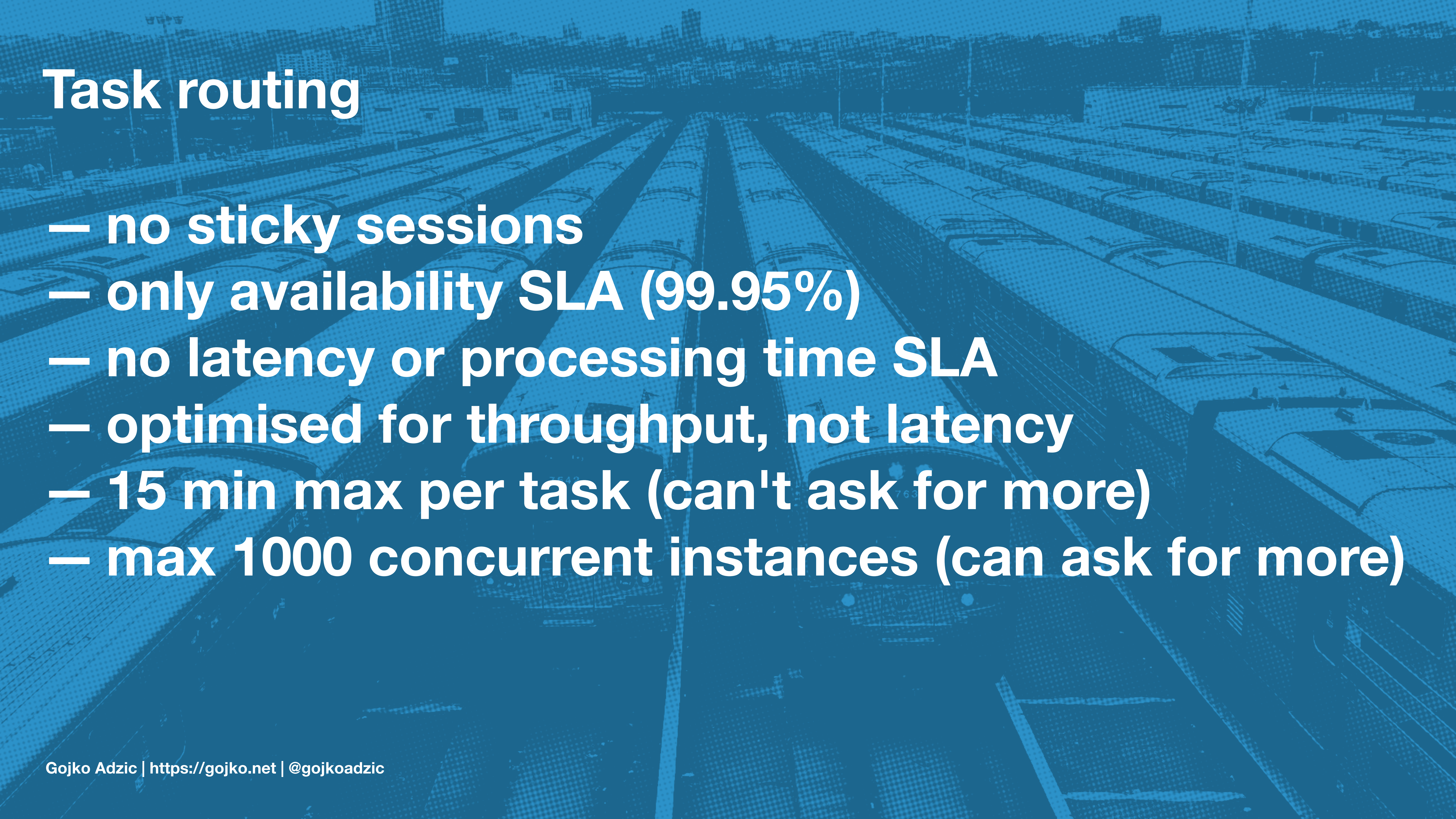
Updating/Consistency

— Use aliases for wiring functions to event sources

— Send data aggregates to functions

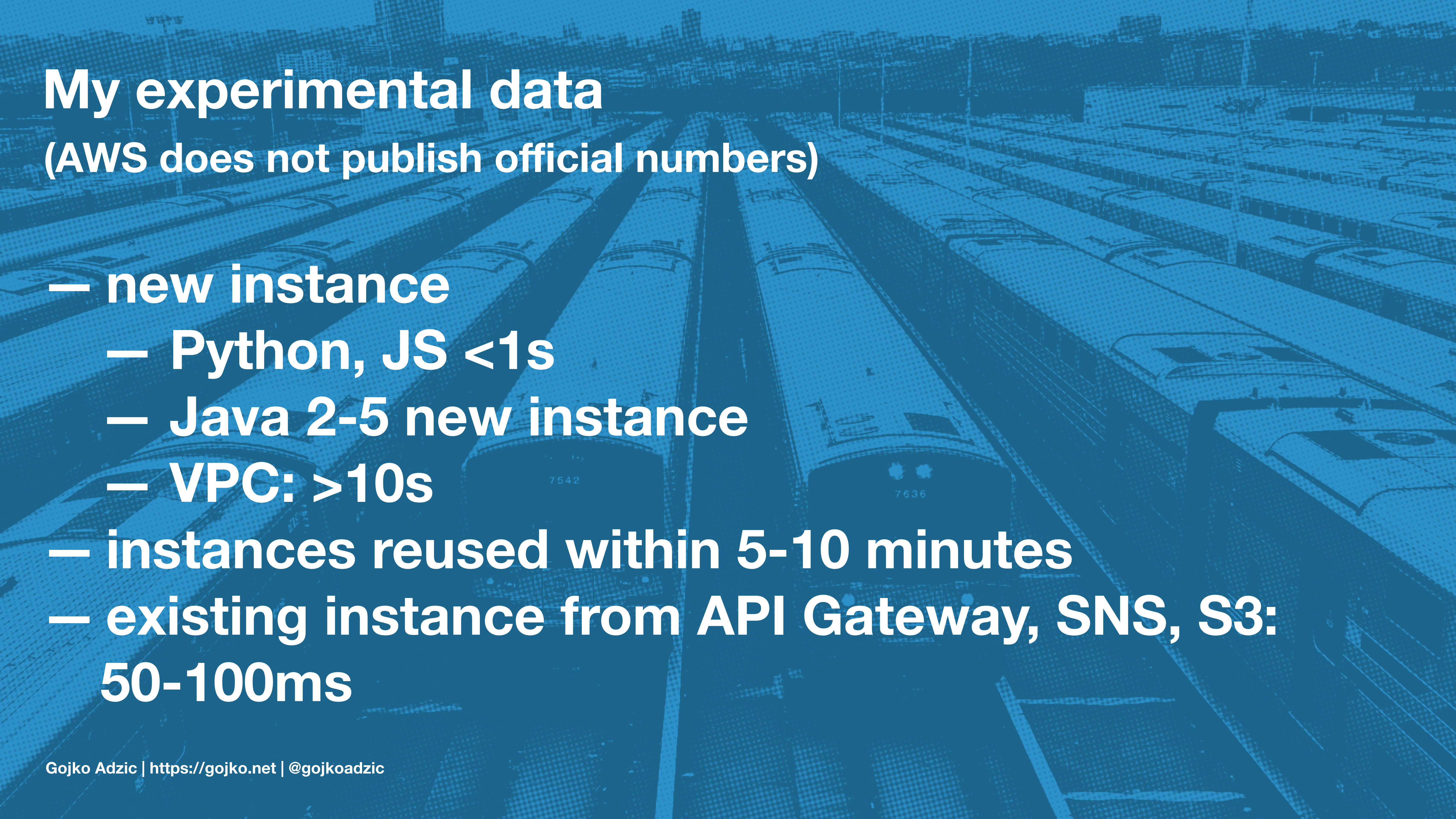
— Write format versions into aggregates

— Bundle code that absolutely must be consistent into the same function



Task routing

— no sticky sessions
— only availability SLA (99.95%)
— no latency or processing time SLA
— optimised for throughput, not latency
— 15 min max per task (can't ask for more)
— max 1000 concurrent instances (can ask for more)



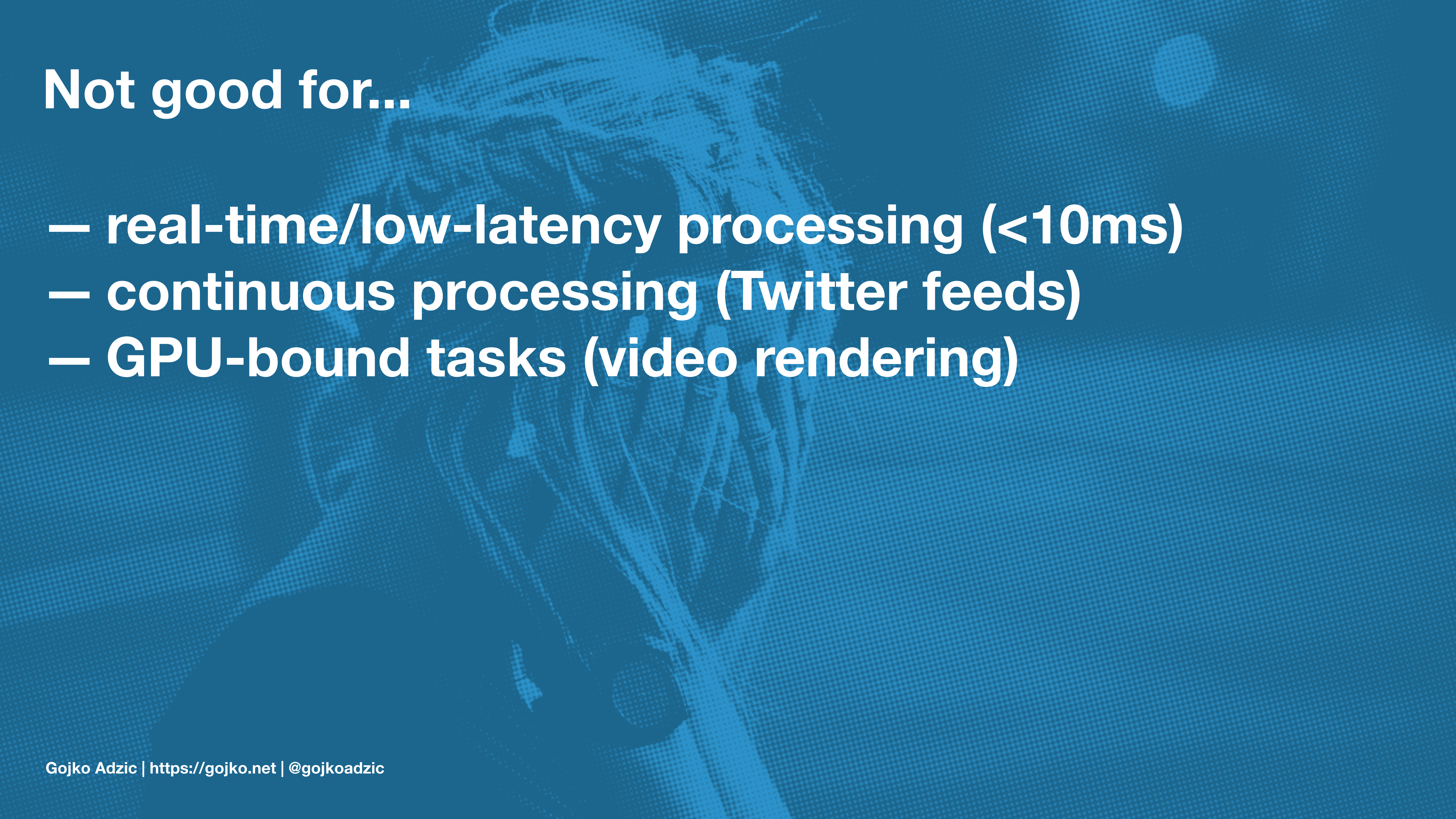
My experimental data

(AWS does not publish official numbers)

- new instance
  - Python, JS <1s
  - Java 2-5 new instance
  - VPC: >10s
- instances reused within 5-10 minutes
- existing instance from API Gateway, SNS, S3: 50-100ms

Great for...

- — HTTP API
- — Image conversions
- — Payment processing
- — Reporting



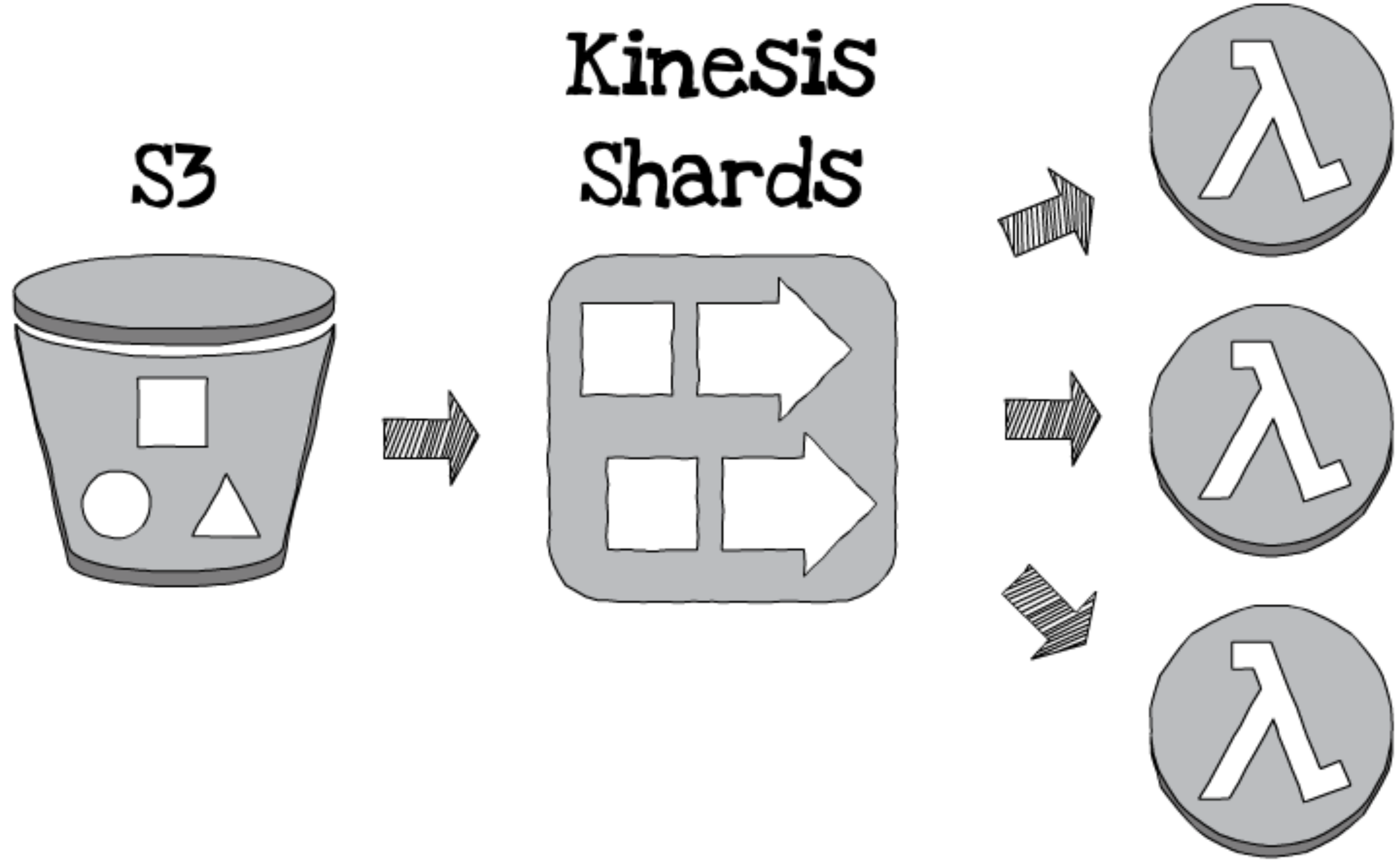
Not good for...

- — real-time/low-latency processing (<10ms)
- — continuous processing (Twitter feeds)
- — GPU-bound tasks (video rendering)

Preventing abuse

```
ConvertFileFunction:
  Type: AWS::Serverless::Function
  Properties:
    ReservedConcurrentExecutions: 10
WebApi:
  Type: AWS::Serverless::Api
  Properties:
    ThrottlingBurstLimit: 20
    ThrottlingRateLimit: 10
```

Controlled parallelisation



Not stateless, but transient

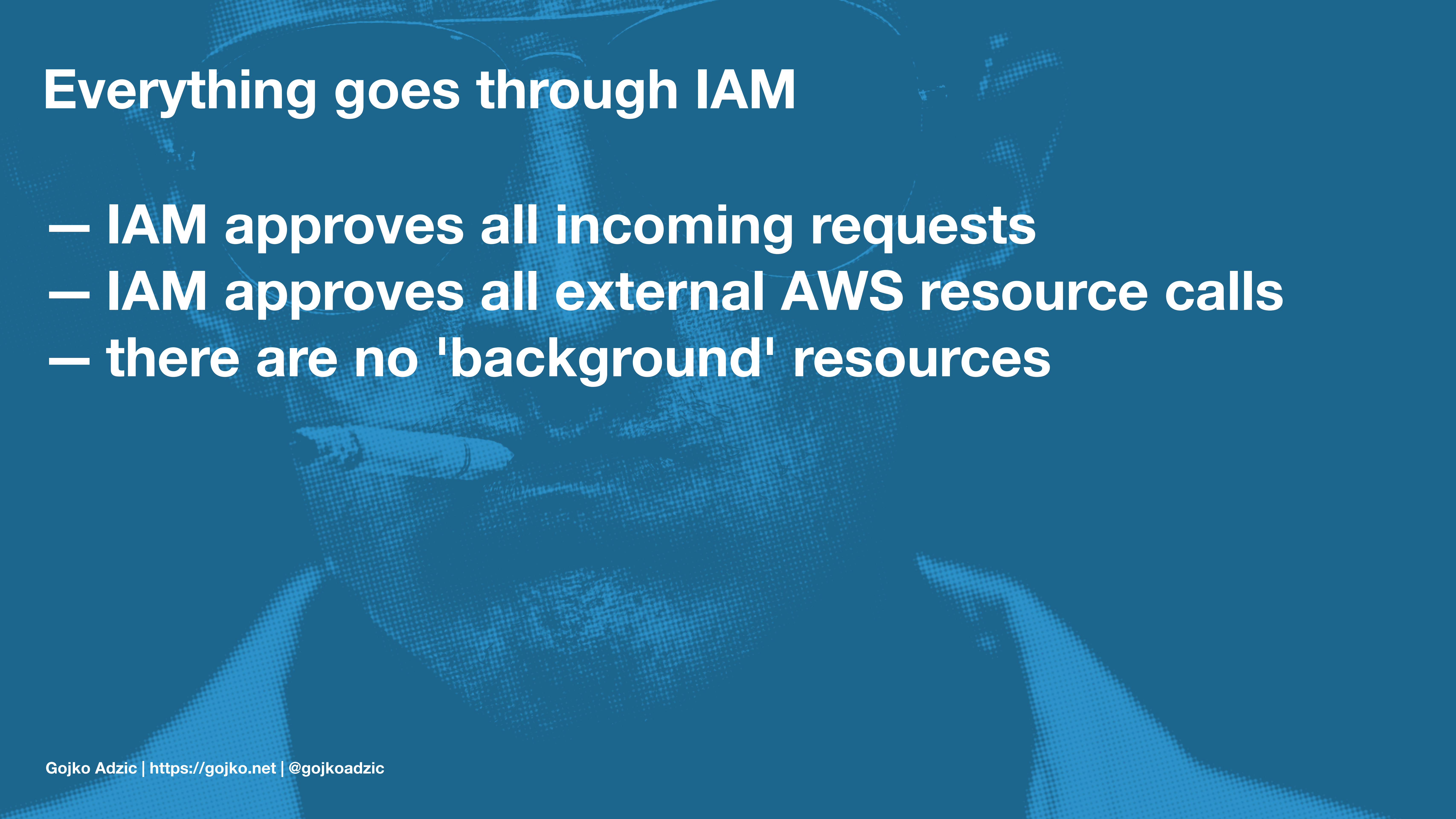
- Instance memory preserved between requests
- `/tmp` **512 MB**

Don't keep user data in Lambda functions

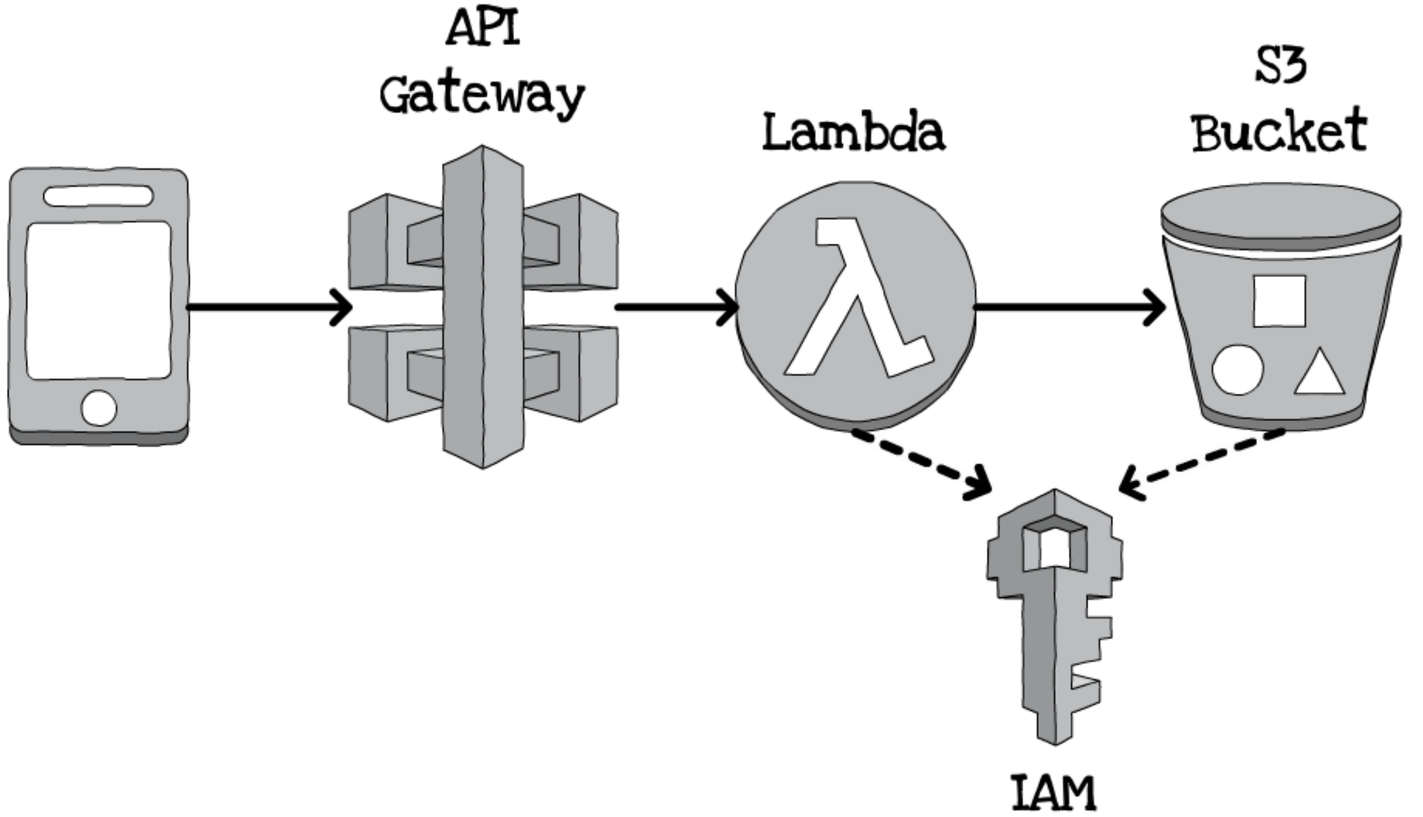
- S3
- DynamoDB
- AppSync

Everything goes through IAM

— IAM approves all incoming requests
— IAM approves all external AWS resource calls
— there are no 'background' resources

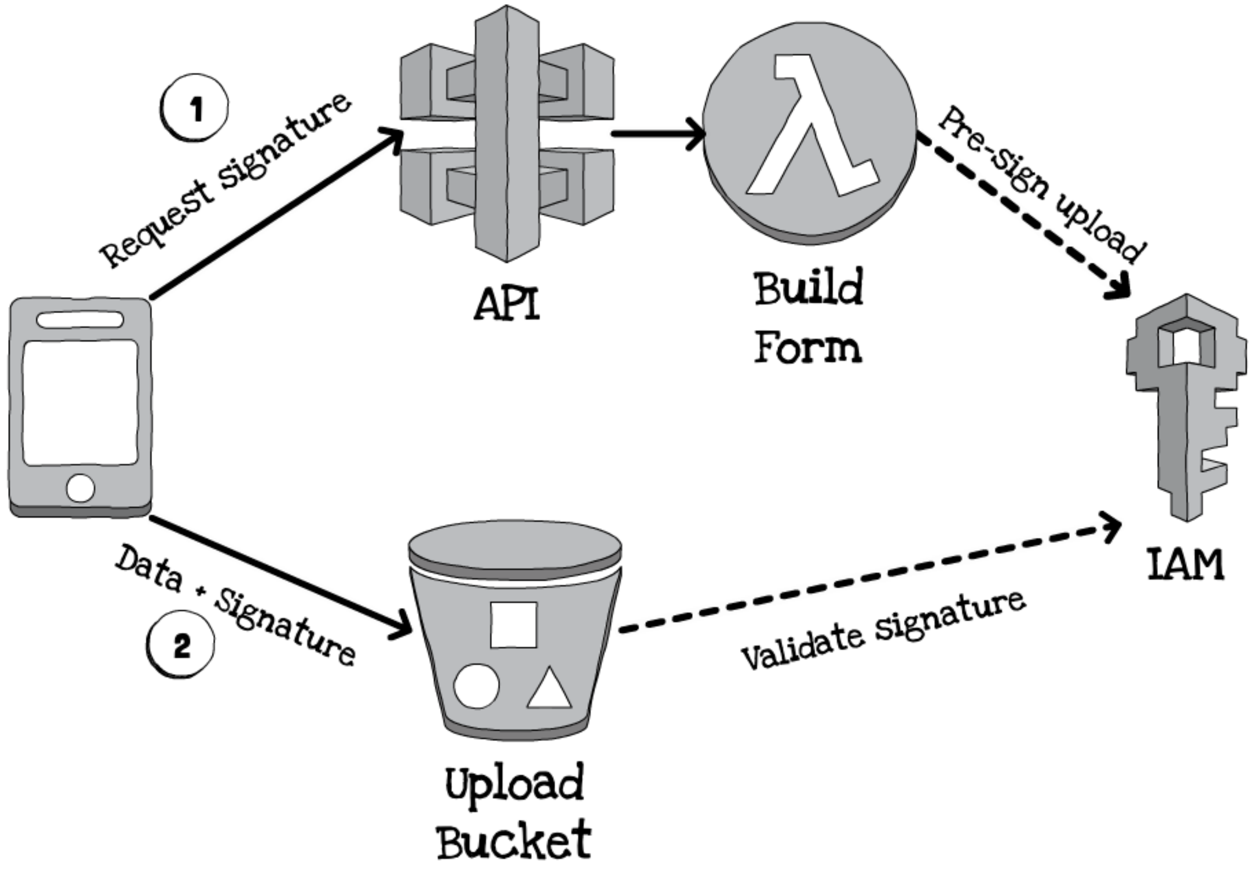


What's it doing here?



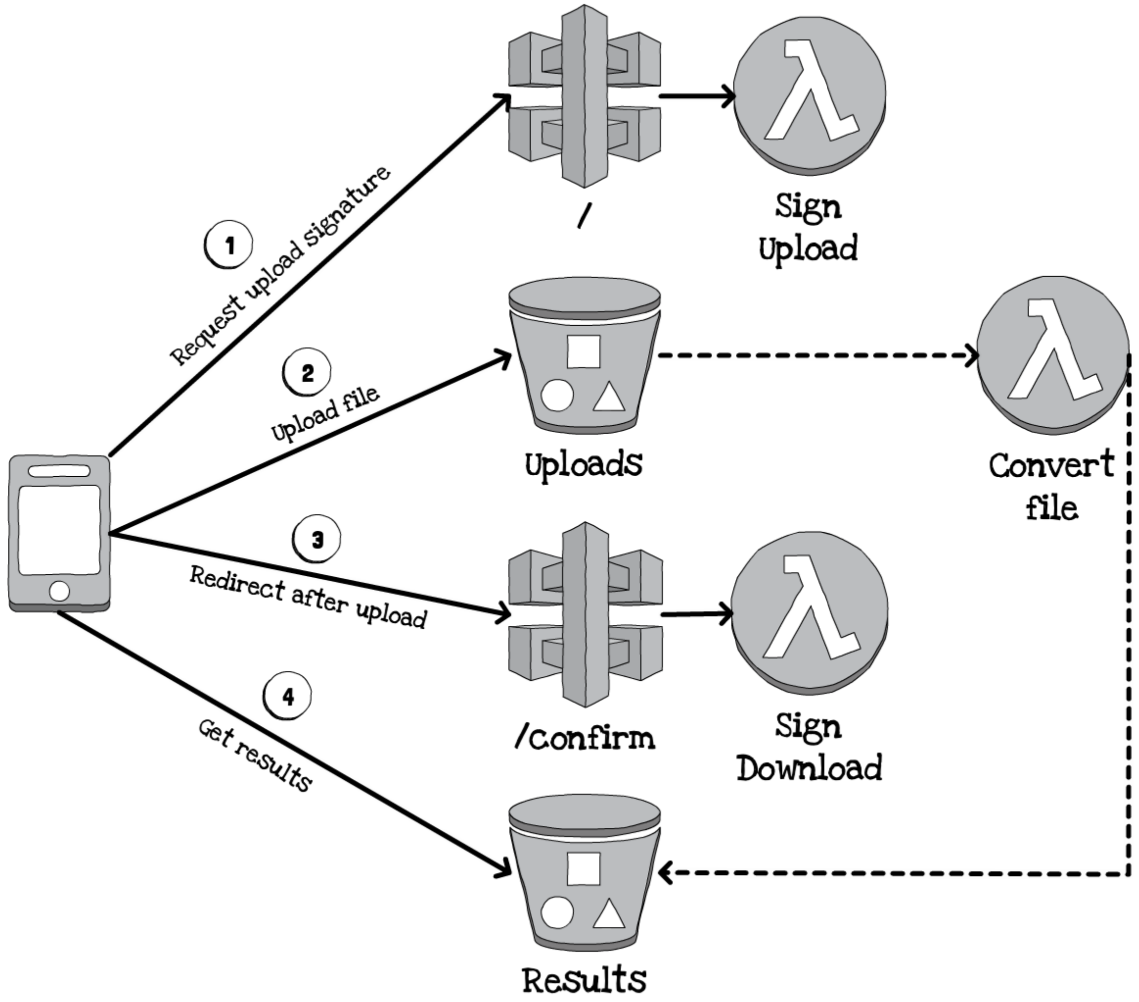
Don't use Lambda as a gatekeeper

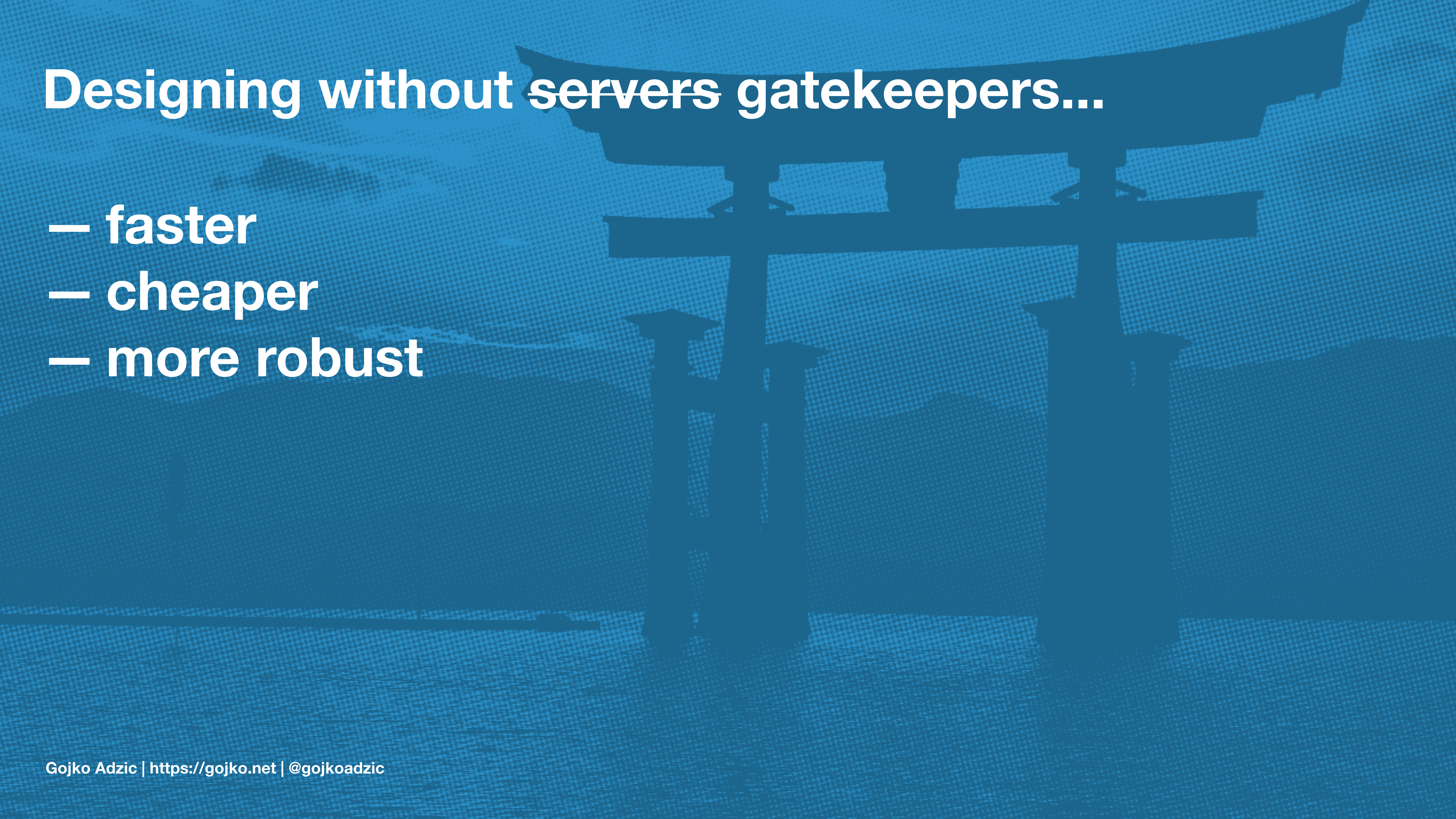
Let client devices talk directly to resources



Don't use Lambda for orchestration

Use platform events + client-side workflows





Designing without ~~servers~~ gatekeepers...

— faster
— cheaper
— more robust

RUNNING SERVERLESS

Gojko Adzic

RUNNING
SERVERLESS
Gojko Adzic
introduction to aws lambda and the serverless application model

http://leanpub.com/running-serverless/c/gotoams

50% off for the next 24 hours