AWS re:Invent

NOV. 29 – DEC. 3, 2021 | LAS VEGAS, NV

COM201

Productizing a serverless MVP

Ivonne Roberts (she/her)
Software Architect, AWS Serverless Hero
Bill.com

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Minimum viable product

What is a minimum viable product (MVP)?

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

“A minimum viable product (MVP) is a version of a product with just enough features to be usable by early customers who can then provide feedback for future product development.”

Techopedia

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Value of an MVP

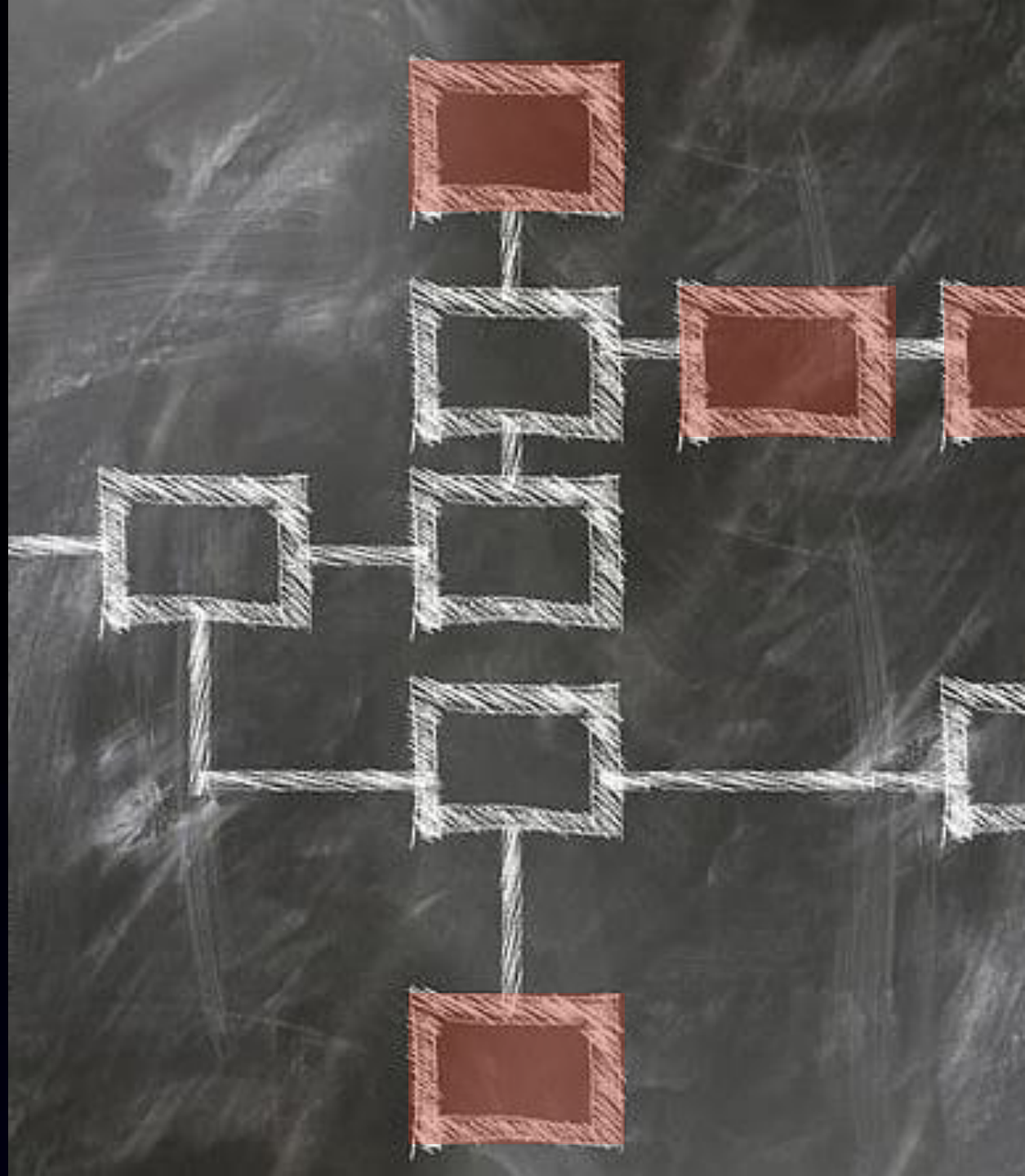
- Allows you to fail fast
- Shortens the feedback loop
- Build what the customer needs



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

What is missing in an MVP?

- Product features
- Coding best practices
- Manually created resources
- Overly provisioned capacity



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

You built an MVP!
Now what?

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Agenda

Development

Security

Deployment

Observability

Disaster recovery

Demo

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Development

How do you design your application?

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

How do you organize your code?

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

“Both the user-side and the server-side problems actually are caused by the same error in design and programming – the entanglement between the business logic and the interaction with external entities.”

Alistair Cockburn

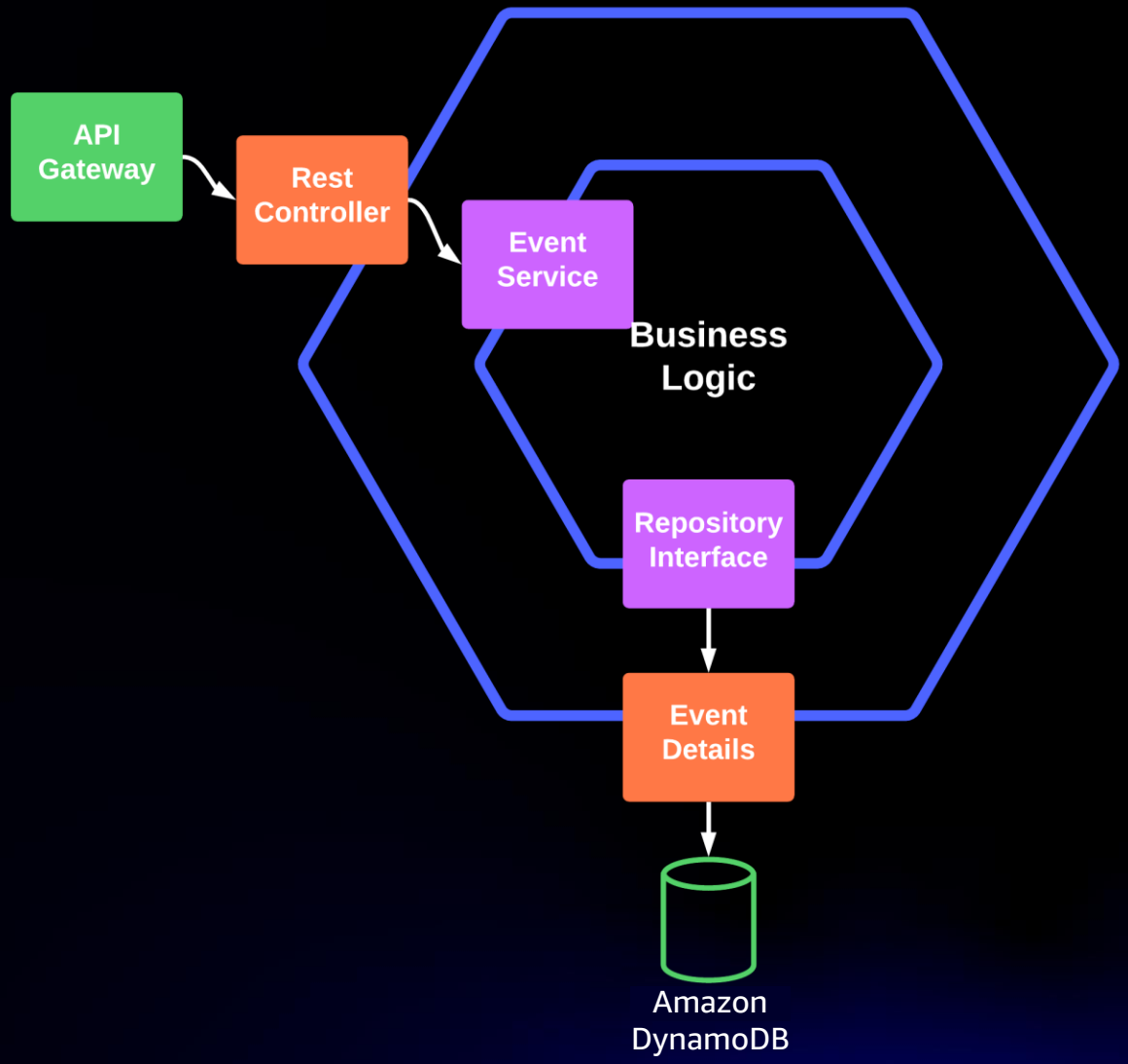
Co-author of *The Agile Manifesto*, creator of hexagonal architecture

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Software design patterns – Directories

Hexagonal architecture

- API (inbound adapters)
- Adapter (outbound third-party adapters)
- Application (ports)
- Domain (outbound ports/adapters)
- Framework



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Software design patterns – Classes and methods

- Write short, simple units of code
- Keep unit interfaces small
- Couple components loosely
- Separate concerns
- Write code once
- Automate tests

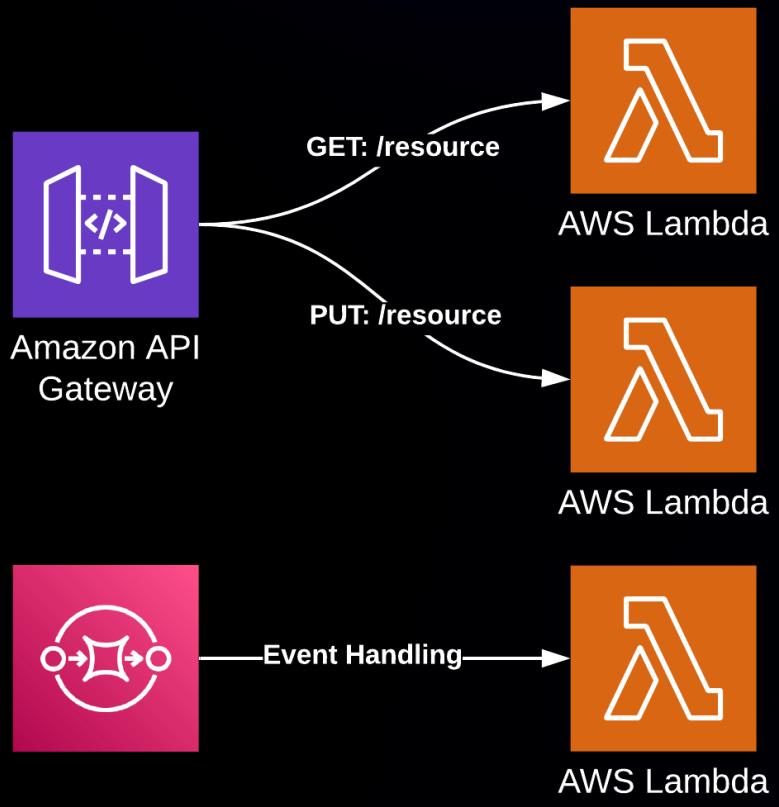


© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

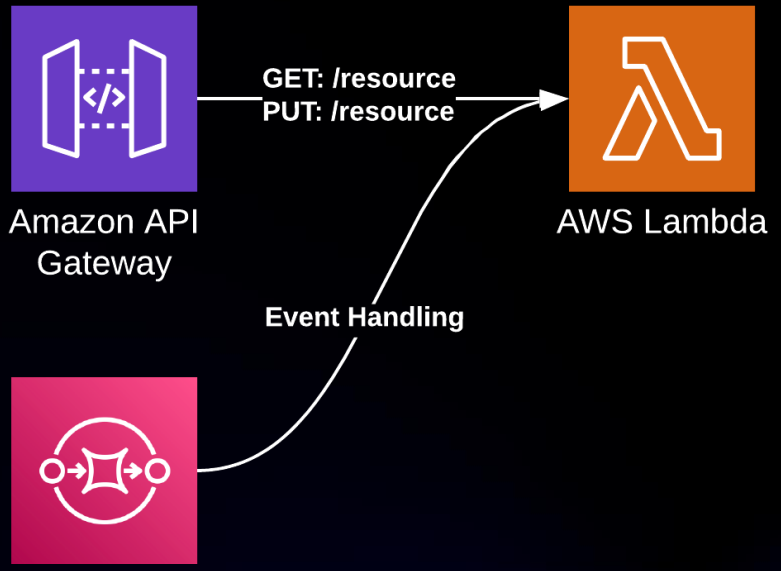
Image by RAEng_Publications from Pixabay

Architecture design patterns

Single function



Lambdalith



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Security

How do you harden your application?

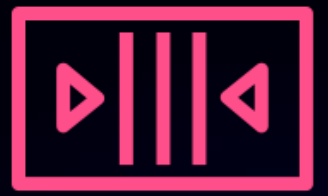
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Security on serverless – Lambda

- Ensure least privileged
- Reserved concurrency
- Dead-letter queue
- Secure secrets
- Input/request validation









© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Security on serverless – API Gateway

- Throttling requests
- Authorizer (Amazon Cognito or Lambda authorizer)
- Resource policies
- Cross-origin resource sharing (CORS)
- AWS WAF



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Deployment

How do you deploy your application?

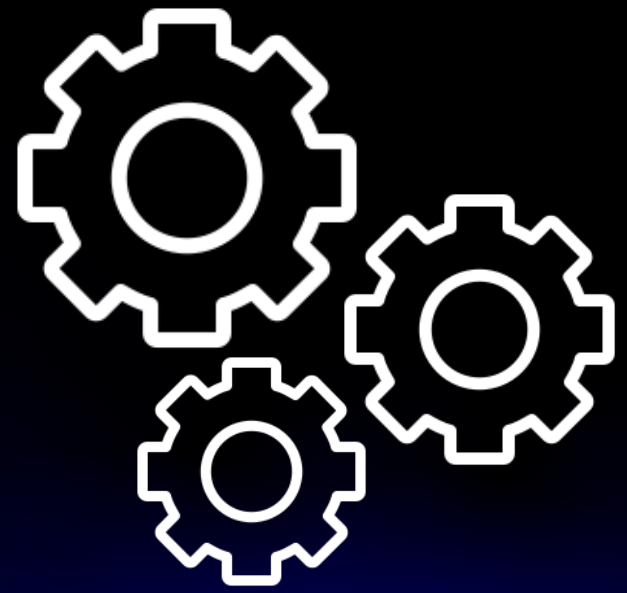
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

How do you deploy your application?

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

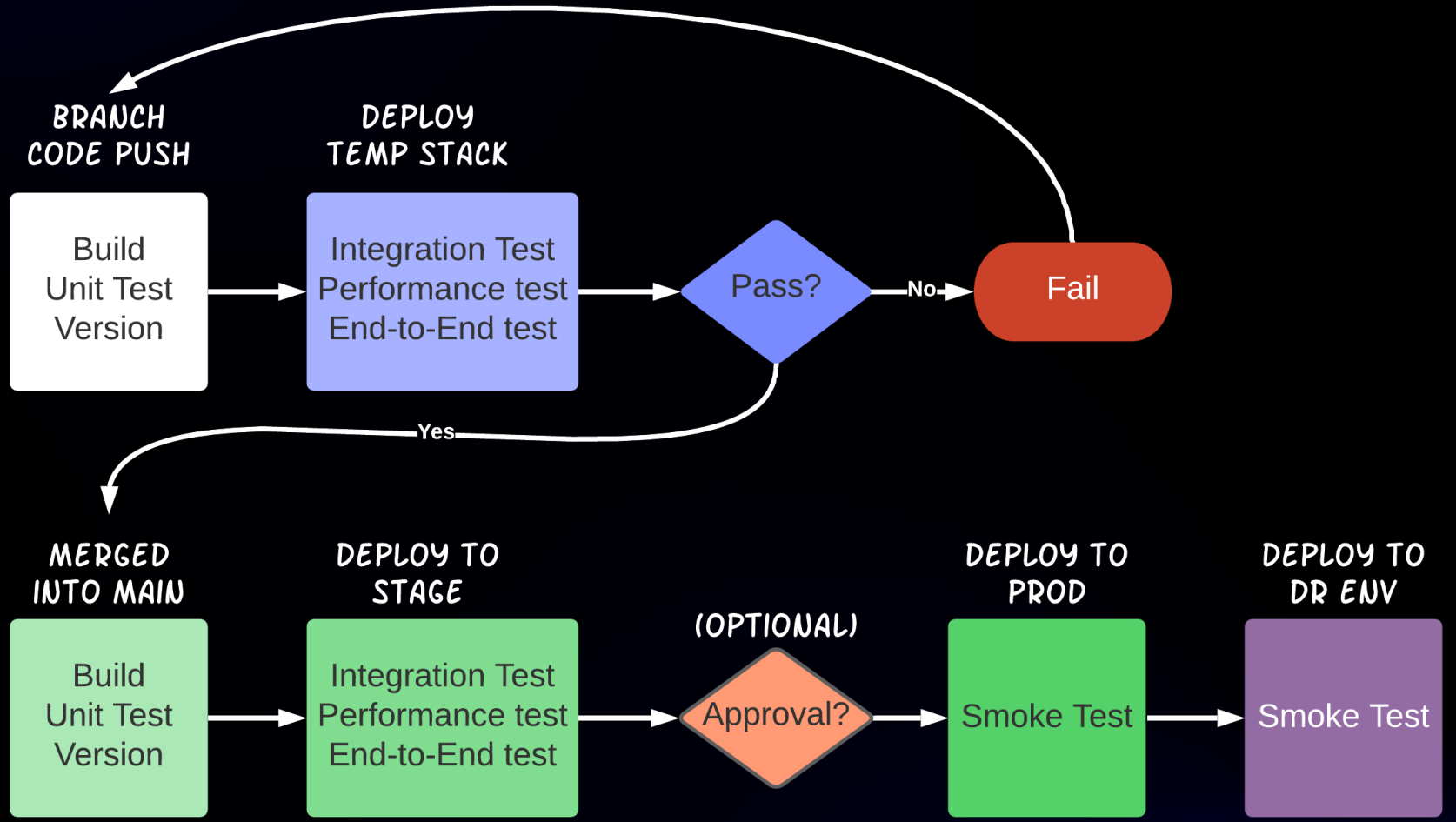
Deploying your application – Infrastructure as code

- Why?
  - Cost – replaces manual steps with an automated process
  - Speed – automation allows faster deployment
  - Risk – repeatable deployment
- Tools
  - AWS SAM
  - AWS CDK
  - AWS CloudFormation
  - Terraform



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Deploying your application – CI/CD



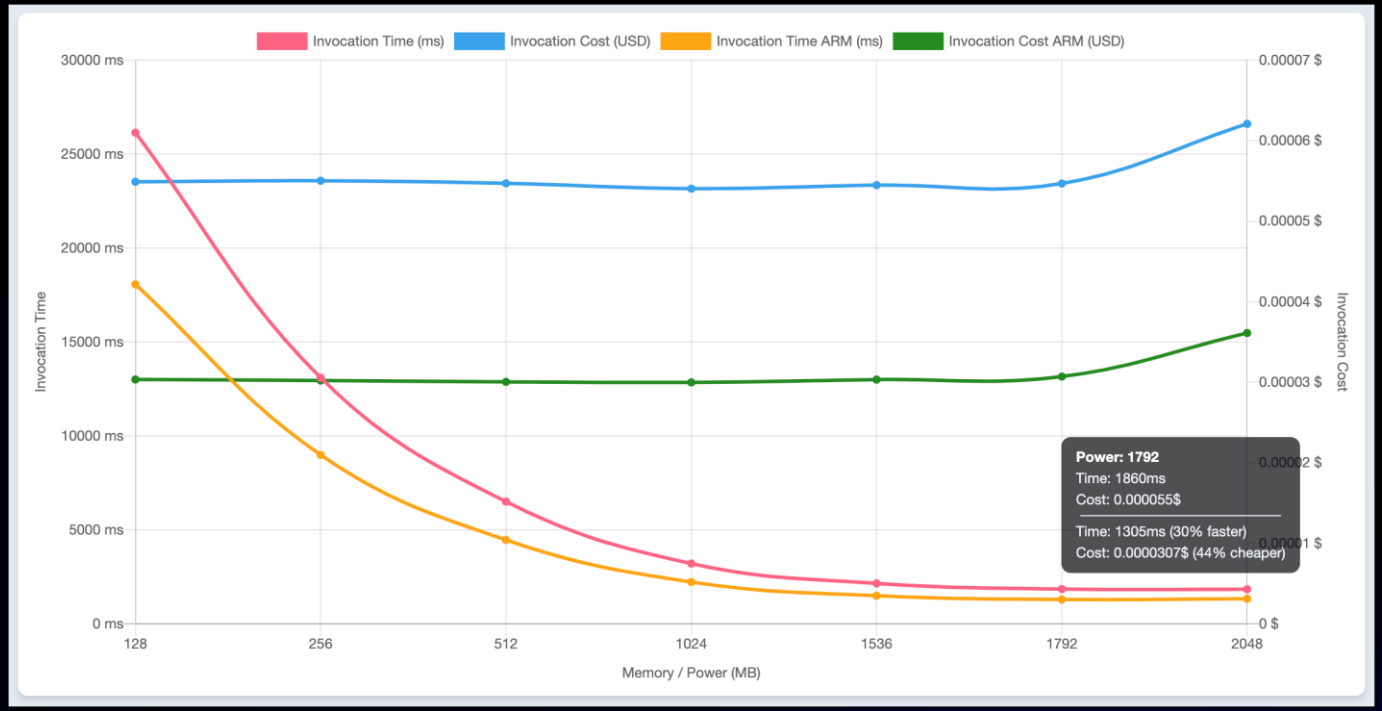
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

How do you optimize cost versus performance?

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Cost versus performance – AWS Lambda

- Automated way to both record and replay request in a staged environment
- Measure
  - Analysis of p99 in logs
  - AWS Lambda Power Tuning
  - AWS Cost Optimizer



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Observability

How do you know what your application is doing?

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

> **“Observability is for running and understanding *your* code – the code you write, change, and ship every day; the code that solves your core business problems.”**

Charity Majors

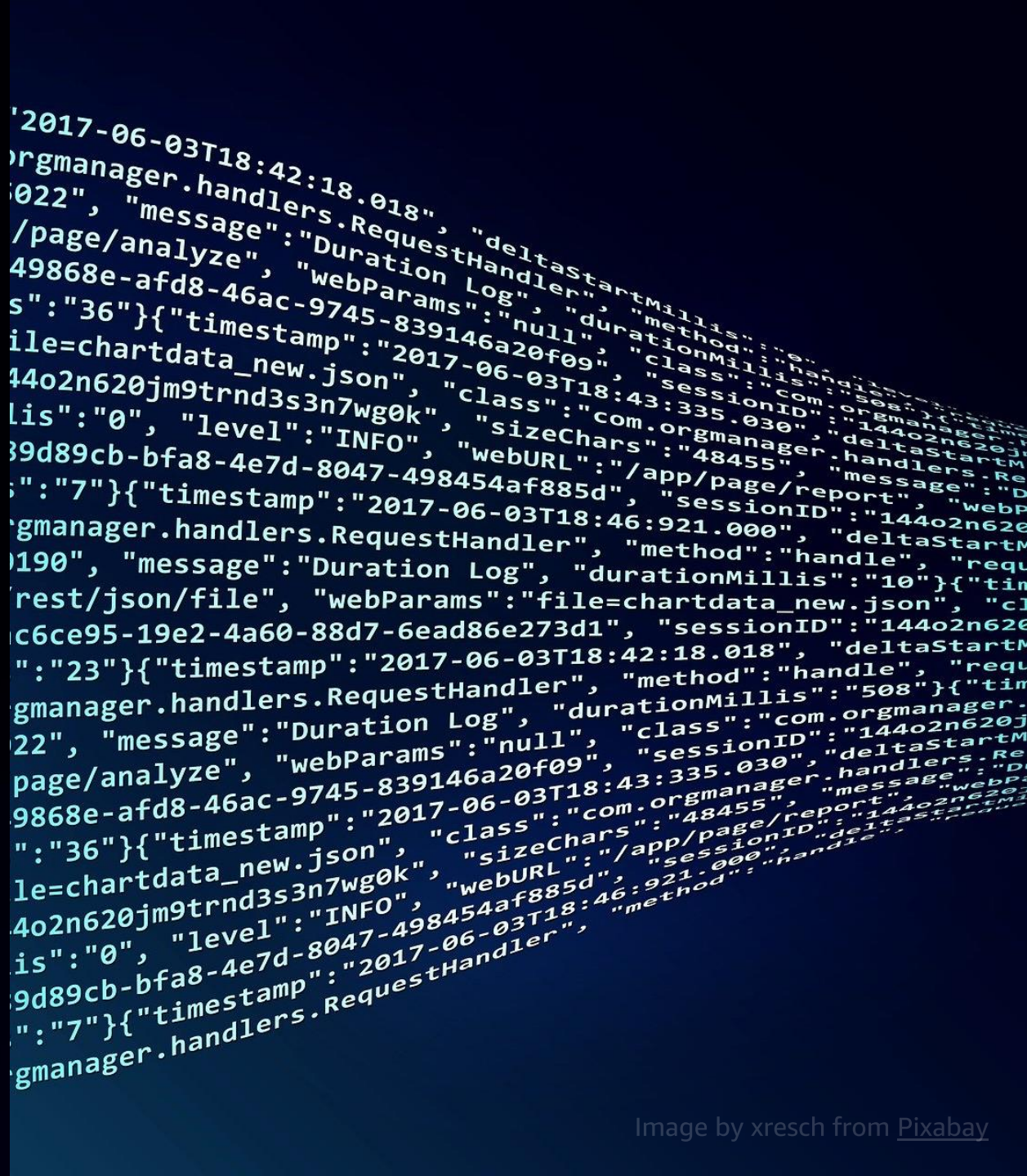
Cofounder/CTO at honeycomb.io

aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Observability – Logging

- What's important?
  - RED dashboard
    - Rate of requests
    - Errors
    - Delay
  - Enough information to debug errors
- Centralized – searchable/accessible
  - Amazon CloudWatch, Splunk, Datadog, New Relic, etc.



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Image by xresch from Pixabay

Observability – Tracing

- See a request traverse microservices and resources
- See the performance for each resource
- Slice and dice on metadata
- Find and debug point of failures



aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Image by Cadop from Pixabay

Disaster recovery

What happens when your application is down?

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

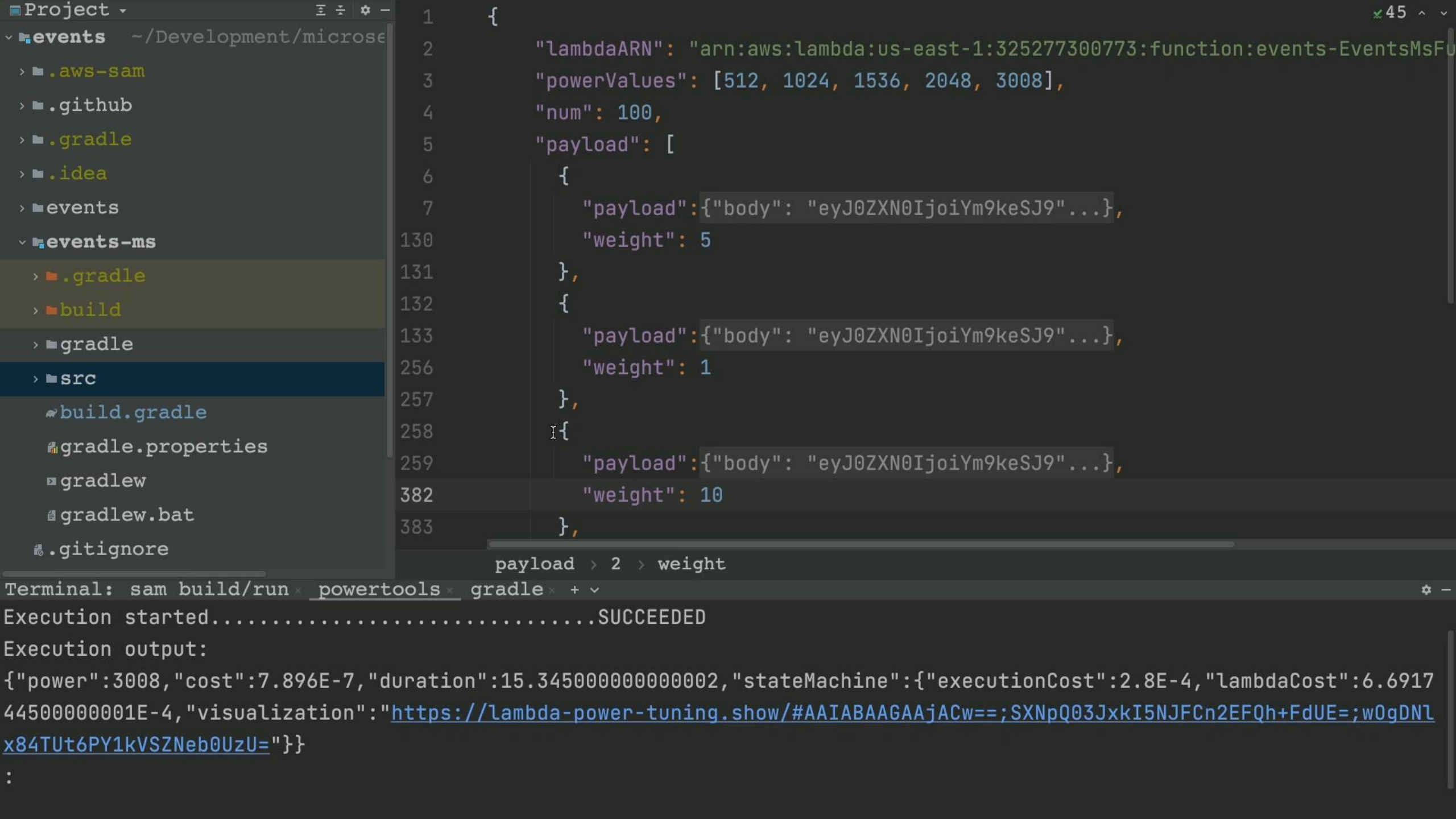
Security on serverless – Disaster recovery

- Define SLAs
  - Recovery time objective (RTO)
  - Recovery point objective (RPO)
- Backup/restore policy for data
- Deployment to a DR site or region
- Automate recovery



Image by Comfreak from Pixabay

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.


Project
events ~/Development/microse
.aws-sam
.github
.gradle
.idea
events
events-ms
.gradle
build
gradle
src
build.gradle
gradle.properties
gradlew
gradlew.bat
.gitignore
{
"lambdaARN": "arn:aws:lambda:us-east-1:325277300773:function:events-EventsMsFu
"powerValues": [512, 1024, 1536, 2048, 3008],
"num": 100,
"payload": [
{
"payload":{"body": "eyJ0ZXN0IjoiYm9keSJ9"...},
"weight": 5
},
{
"payload":{"body": "eyJ0ZXN0IjoiYm9keSJ9"...},
"weight": 1
},
{
"payload":{"body": "eyJ0ZXN0IjoiYm9keSJ9"...},
"weight": 10
},
payload › 2 › weight
Terminal: sam build/run powertools gradle
Execution started.....SUCCEEDED
Execution output:
{"power":3008,"cost":7.896E-7,"duration":15.345000000000002,"stateMachine":{"executionCost":2.8E-4,"lambdaCost":6.6917
44500000001E-4,"visualization":"https://lambda-power-tuning.show/#AAIABAAGAAjACw==;SXNpQ03JxkI5NJFCn2EFQh+FdUE=;wOgDNl
x84TUt6PY1kVSZNeb0UzU="}}

Conclusion

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Rinse and repeat

- You shouldn’t assume any choice you’ve made to be final
- You should
  - Be constantly re-evaluating/ re-enhancing/refactoring
  - Design for flexibility and maintainability



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Image by Alen from stockvault.net

References



Chris Richardson. *Microservices Patterns*. Manning.



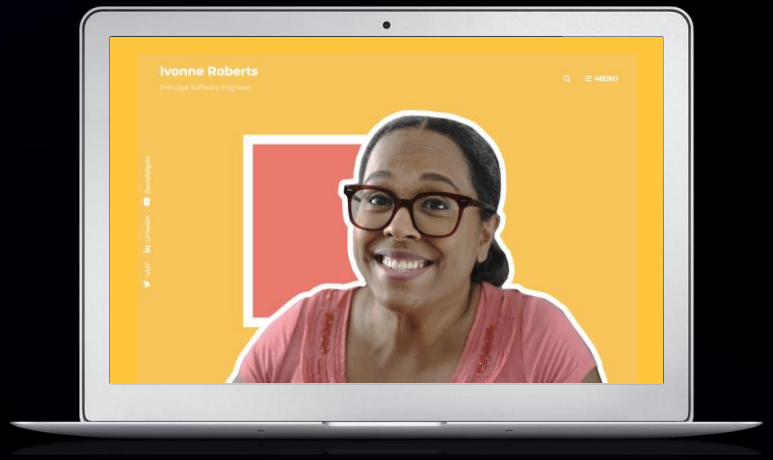
Joost Visser. *Building Maintainable Software,* Java ed. O'Reilly.



Sam Newman. *Monolith to Microservices.* O'Reilly.

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Follow me on the interwebs!



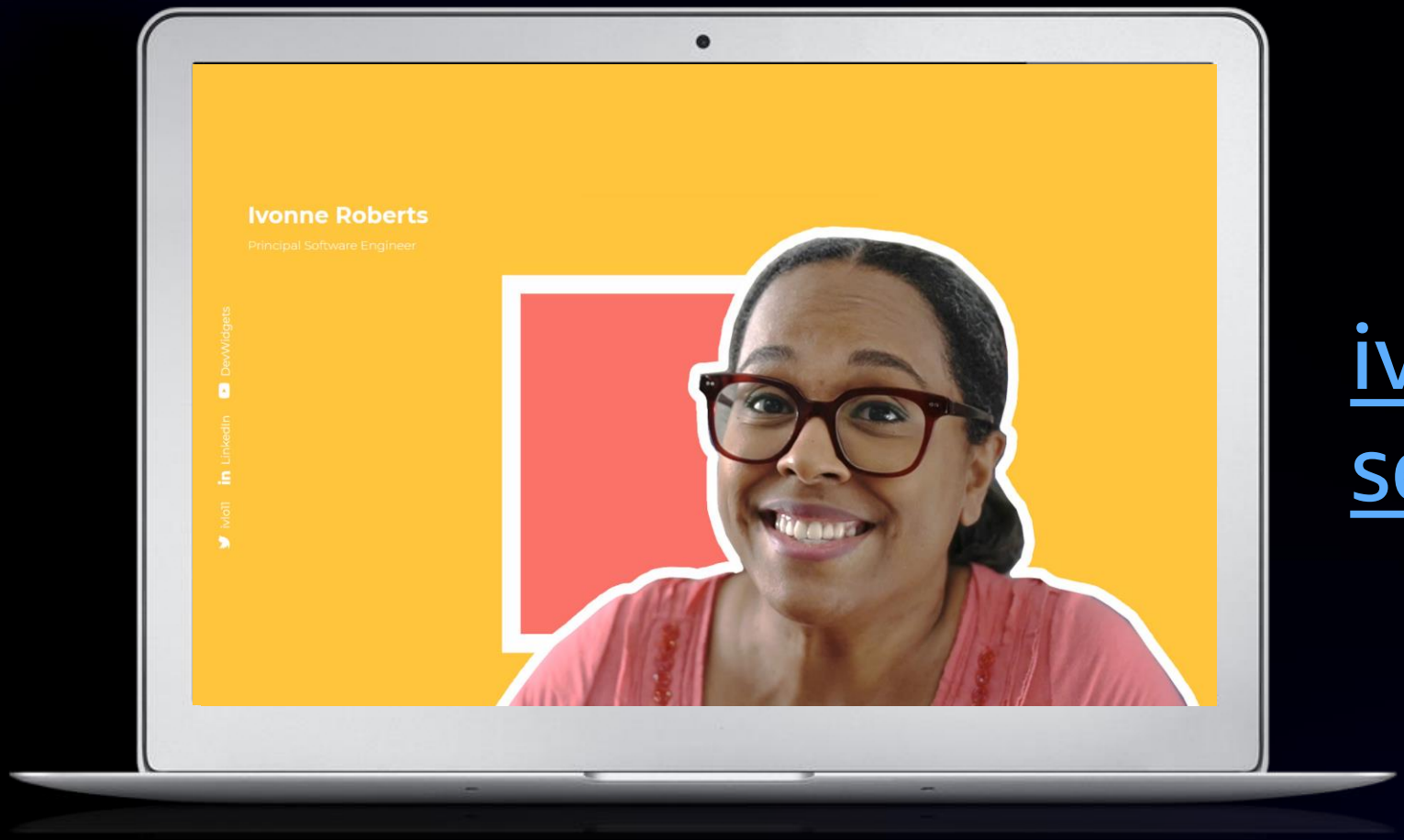
@ivlo11

ivonneroberts.com

Serverless DevWidgets

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

References



ivonneroberts.com/serverless-mvp

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Thank you!

Ivonne Roberts
@ivlo11



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.