AWS re:Invent

DOP409-R

Faster Cryptography in Java with Amazon Corretto Crypto Provider (ACCP)

Petr Praus

Sr. Software Development Engineer

Amazon Web Services

Agenda

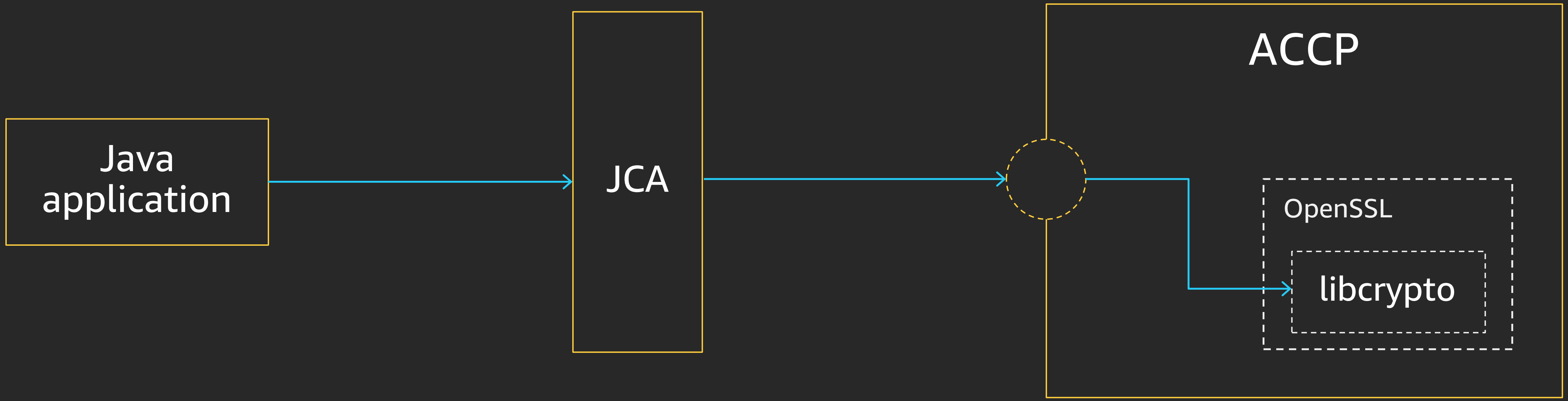
- What is Amazon Corretto
- What is ACCP
- Demo service
- Bonus: Reactive Streams

Amazon Corretto

- OpenJDK distribution from Amazon
- Same open-source license as OpenJDK
- Long-term support at no cost
    - Java 8 until at least June 2023
    - Java 11 until at least August 2024
- Production ready
    - Amazon runs Corretto internally on thousands of production services
- Runs anywhere, not just on AWS

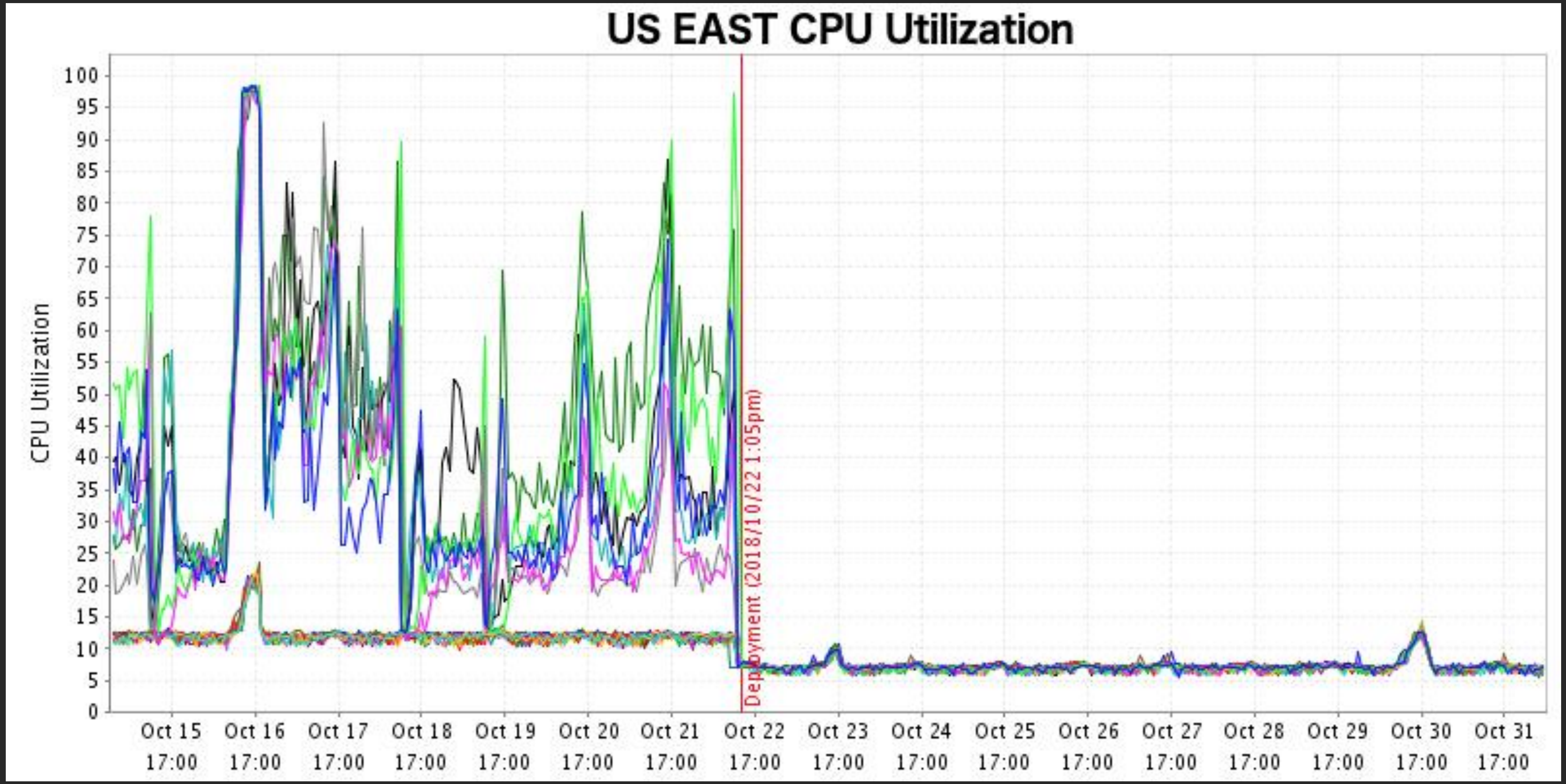
Amazon Corretto Crypto Provider (ACCP)

- Faster cryptography for Java on 64-bit Linux
- Implements Java Cryptography Architecture (JCA) interfaces
- Algorithms implementations from libcrypto (part of OpenSSL)
- AES-GCM, HMAC, SHA, MD5, RSA, DH, ECDSA, ECDH



ACCP

- AWS Snowball on ACCP
  - CPU utilization went from 30% to 85% to stable 8%
  - Doubled data transfer speed



ACCP

How do I use ACCP in my app?

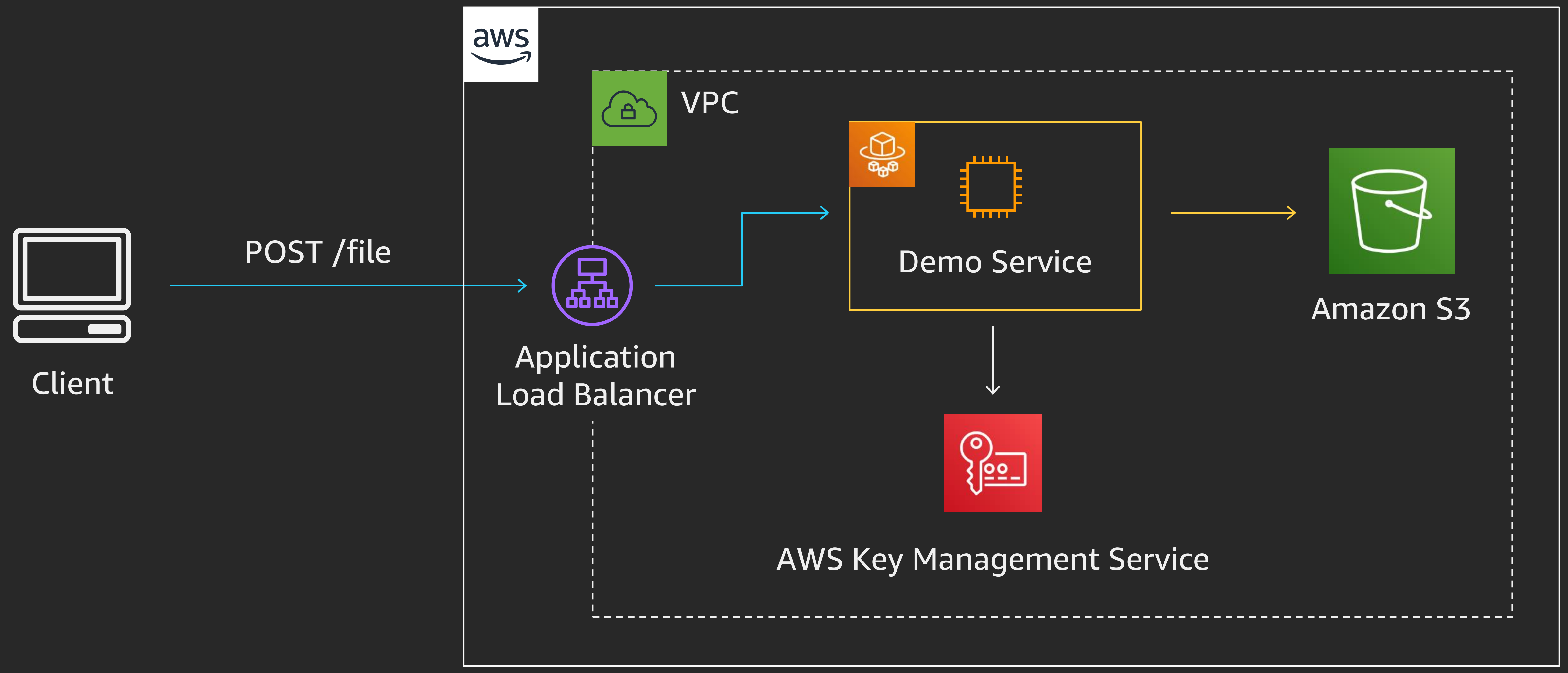
```
// build.gradle
dependencies {
    implementation 'software.amazon.cryptools:AmazonCorrettoCryptoProvider:1.+:linux-x86_64’
}

// Initialization of your app
AmazonCorrettoCryptoProvider.install()
```

Demo service

Demo service

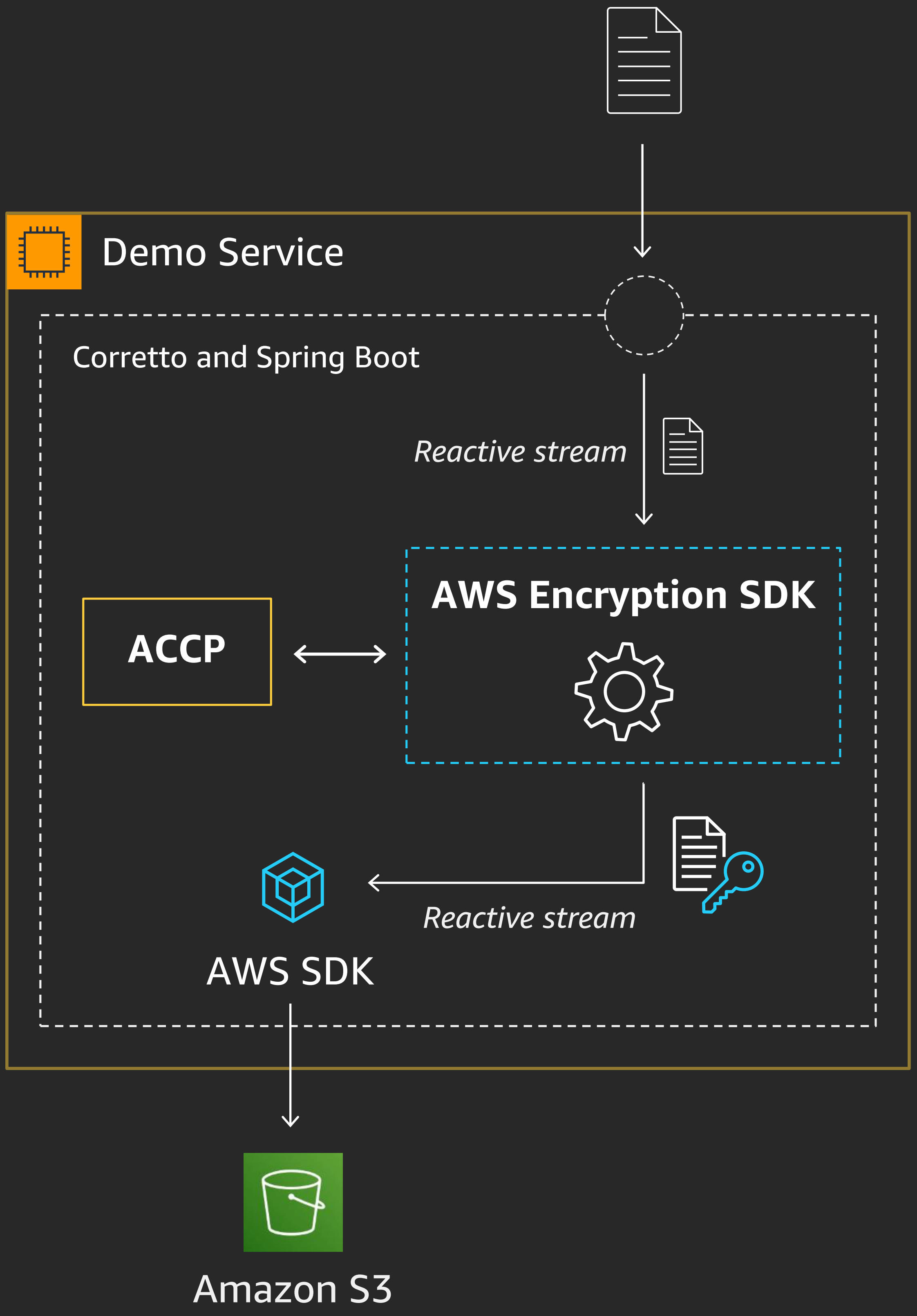
- Encrypts/decrypts *on the fly* using AWS Encryption SDK
- Encrypted files are stored in Amazon S3
- Each file has a unique ID
- Simple REST API
  - `POST /file`
    - Returns file ID and other metadata
  - `GET /file/{id}`
    - Finds file in Amazon S3 and streams the file back

Demo service: AWS architecture



Inside demo service

- Written in Java
- Corretto 8
- Spring Boot
- Reactive Streams and Spring WebFlux
- ACCP
- AWS Encryption SDK for Java



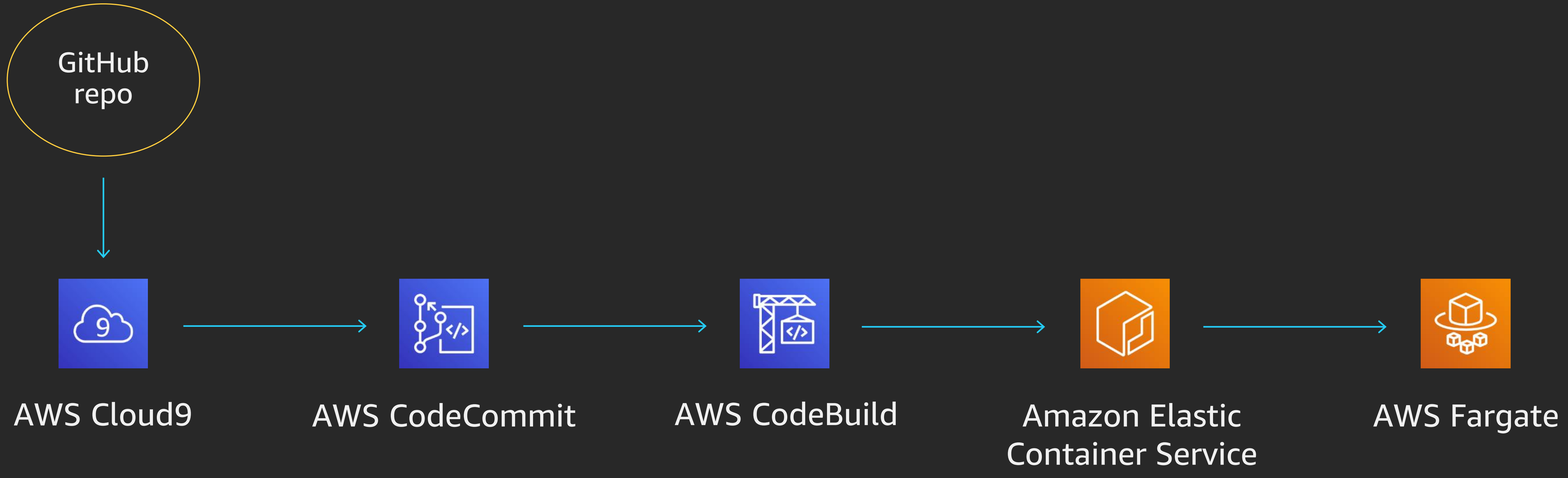
Working with demo service

AWS re:Invent

Demo service on GitHub

https://github.com/aws-samples/faster-cryptography-in-java-with-aws

Demo service: build system



GitHub repo

AWS Cloud9

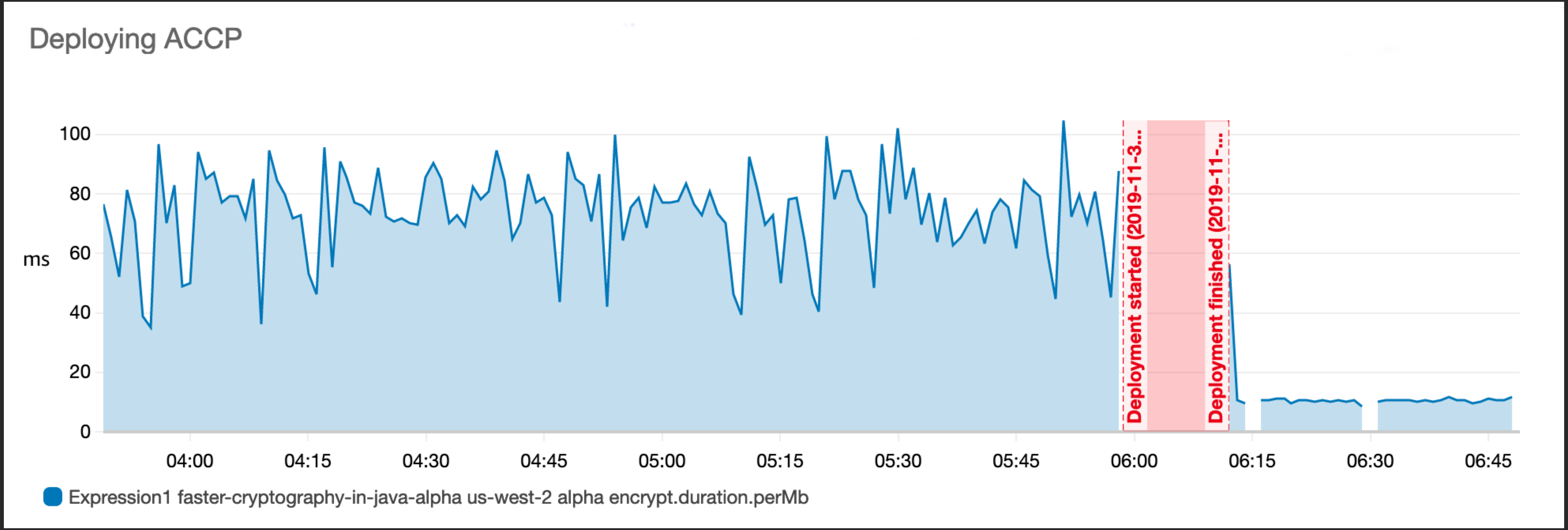
AWS CodeCommit

AWS CodeBuild

Amazon Elastic Container Service

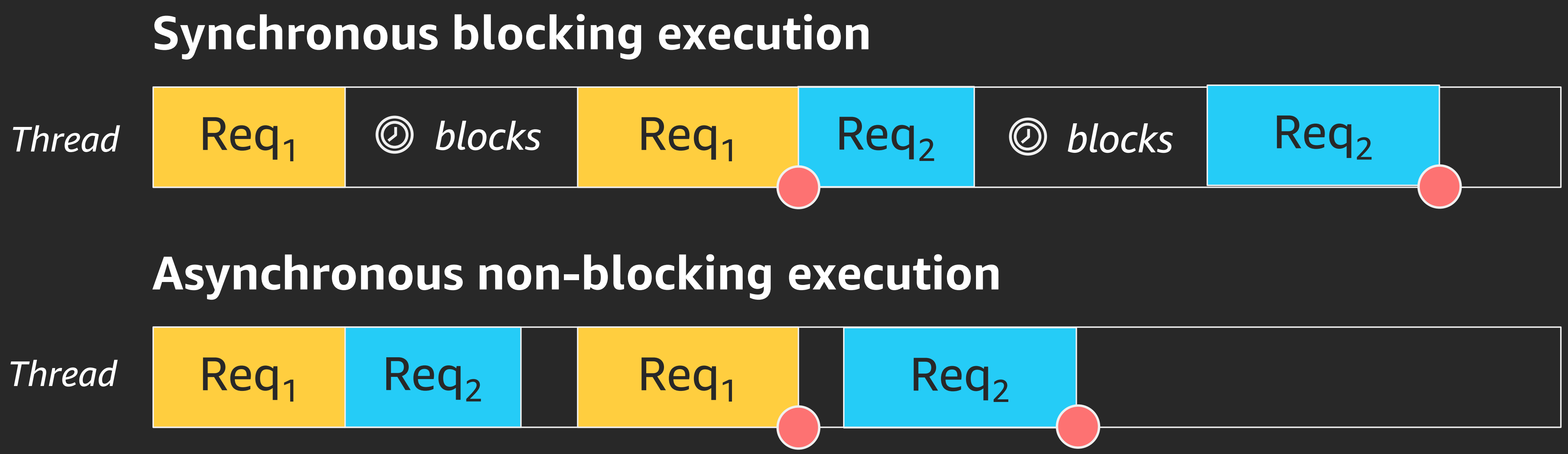
AWS Fargate

Deploying ACCP in demo service



Bonus: reactive streams

- **Problem:** Business logic makes a blocking call to a remote service
- **Typical solution:** Huge thread pools
- **Good solution:** Make the call but do something else until we get a response
- Hard to do in complex business applications (e.g., Spring)
- Reactive Streams brings this concept into mainstream Java development
- Work is divided into many small non-blocking tasks executed on real threads by a scheduler



Related breakouts

SEC401-R – Using the AWS Encryption SDK for multi-master key encryption

Learn DevOps with AWS Training and Certification

Resources created by the experts at AWS to propel your organization and career forward

Take free digital training to learn best practices for developing, deploying, and maintaining applications

Classroom offerings, like DevOps Engineering on AWS, feature AWS expert instructors and hands-on activities

Validate expertise with the **AWS Certified DevOps Engineer - Professional** or **AWS Certified Developer - Associate** exams

Visit aws.amazon.com/training/path-developing/

Thank you!

AWS re:Invent

aws

Please complete the session survey in the mobile app.

AWS re:Invent