
AWS
re:Invent
NOV. 28 – DEC. 2, 2022 | LAS VEGAS, NV

XNT402

Cloud-native .NET applications on AWS

Brian Lewis
Solutions Architect
AWS

Sanjay Gulati
Solutions Architect
AWS

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Agenda

Why build modern applications?

Looking under the covers of a modern .NET reference application

Frontend architecture

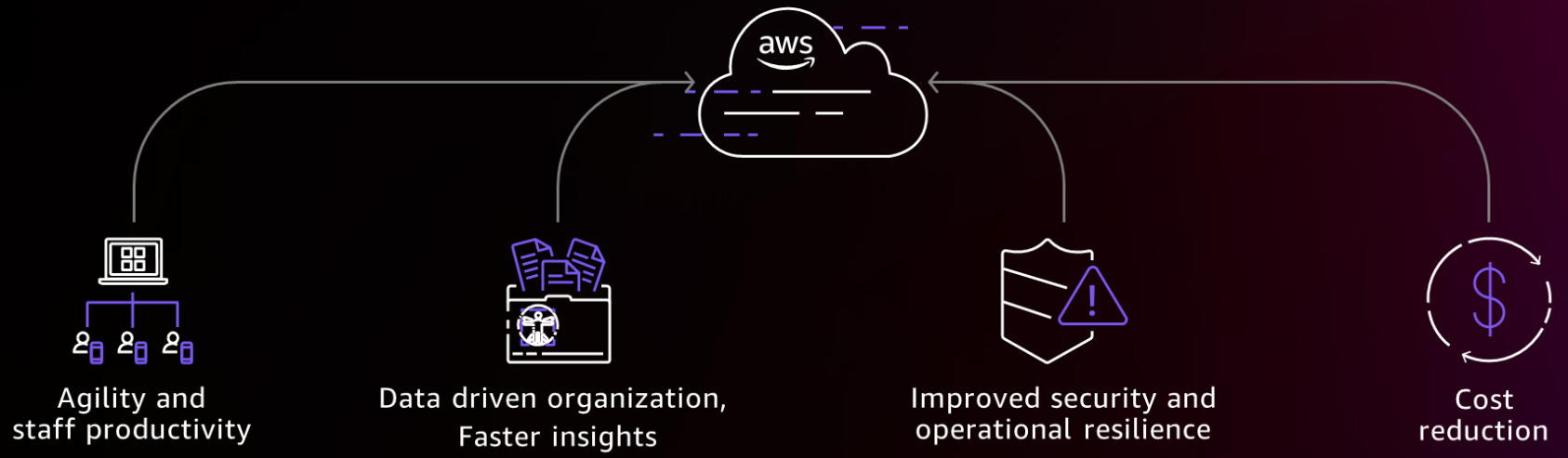
Backend architecture

Troubleshooting in a modern app

Where you can get the code

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Why build a modern application?



Modern applications and infrastructure take advantage of cloud economics (costs) and operating models (scale & agility) to accelerate innovation

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS customer example



At peak consumes 1/3 of US internet bandwidth

Deploys thousands of deployments per day

Self-healing design – Chaos Monkey testing

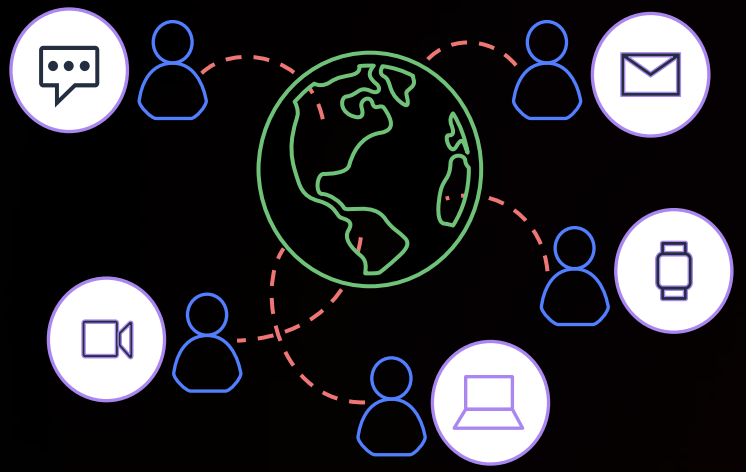
https://aws.amazon.com/solutions/case-studies

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Characteristics of a modern app

.NET VERSION OF A MODERN APP




e-commerce


Media streaming


Social media


Online gaming


Shared economy

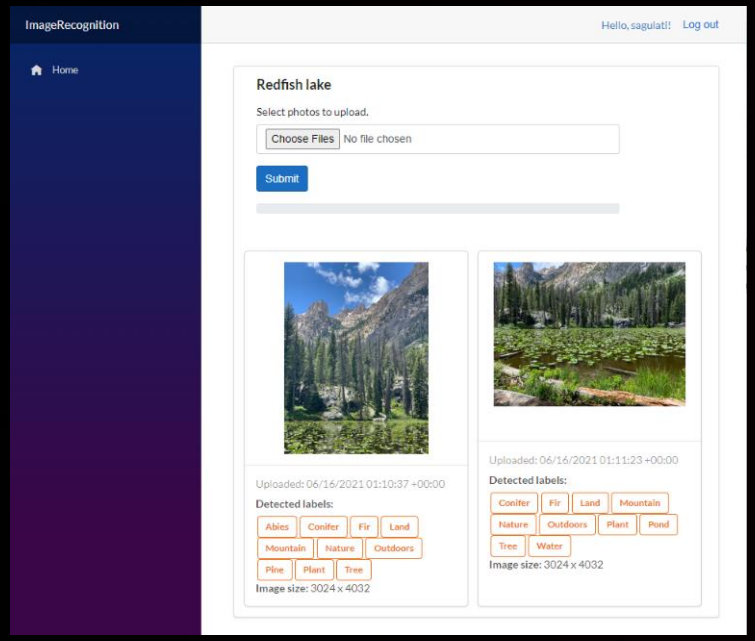
- Use independently scalable microservices (serverless, containers)
- Connect through APIs
- Deliver updates continuously
- Scale globally
- Are fault tolerant
- Manage state and persistence
- Have security built in
- Decoupled data strategy

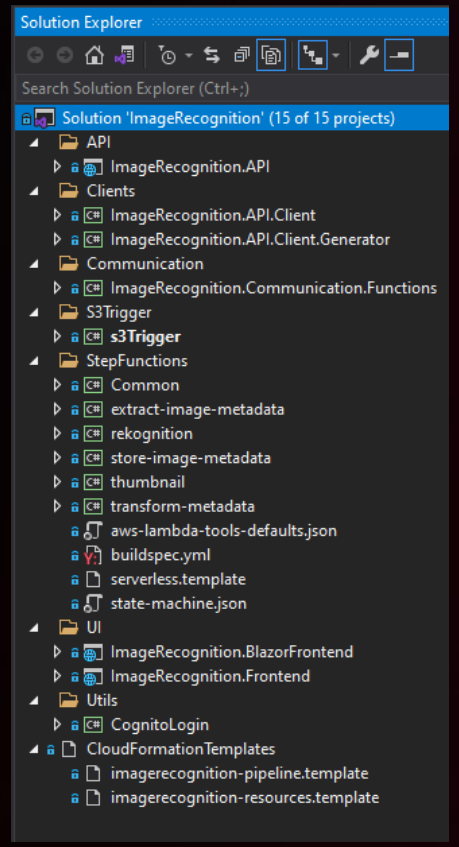
aws

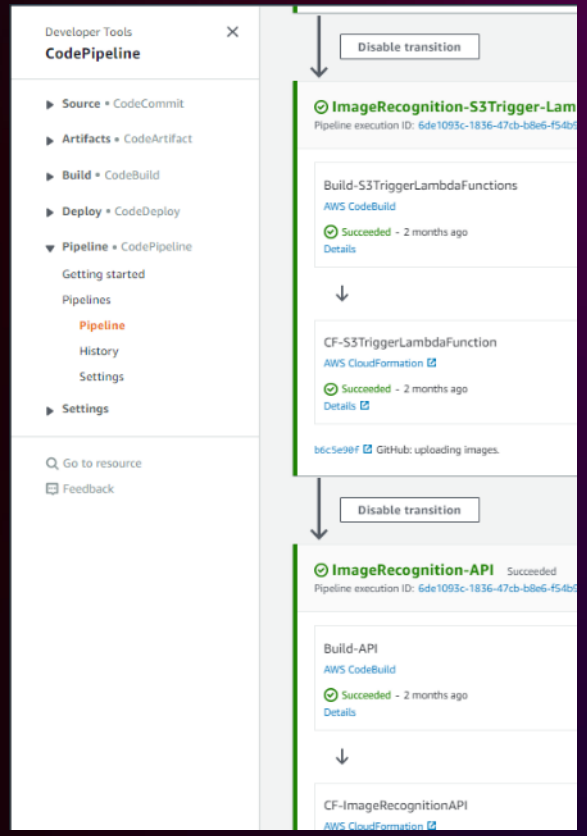
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

The reference application we built

.NET SERVERLESS IMAGE RECOGNITION APP







© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Technologies used in this app



- Amazon Cognito
- Amazon Dynamo DB
- Amazon S3
- Amazon CloudFront
- Amazon API Gateway
- AWS Lambda
- Lambda functions
- AWS X-Ray
- Modern app design
- AWS SDK for .NET
- WebSockets
- CI/CD practice
- Troubleshooting serverless application bugs
- Microsoft Blazor UI (one page applications)

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS tools used to build this app



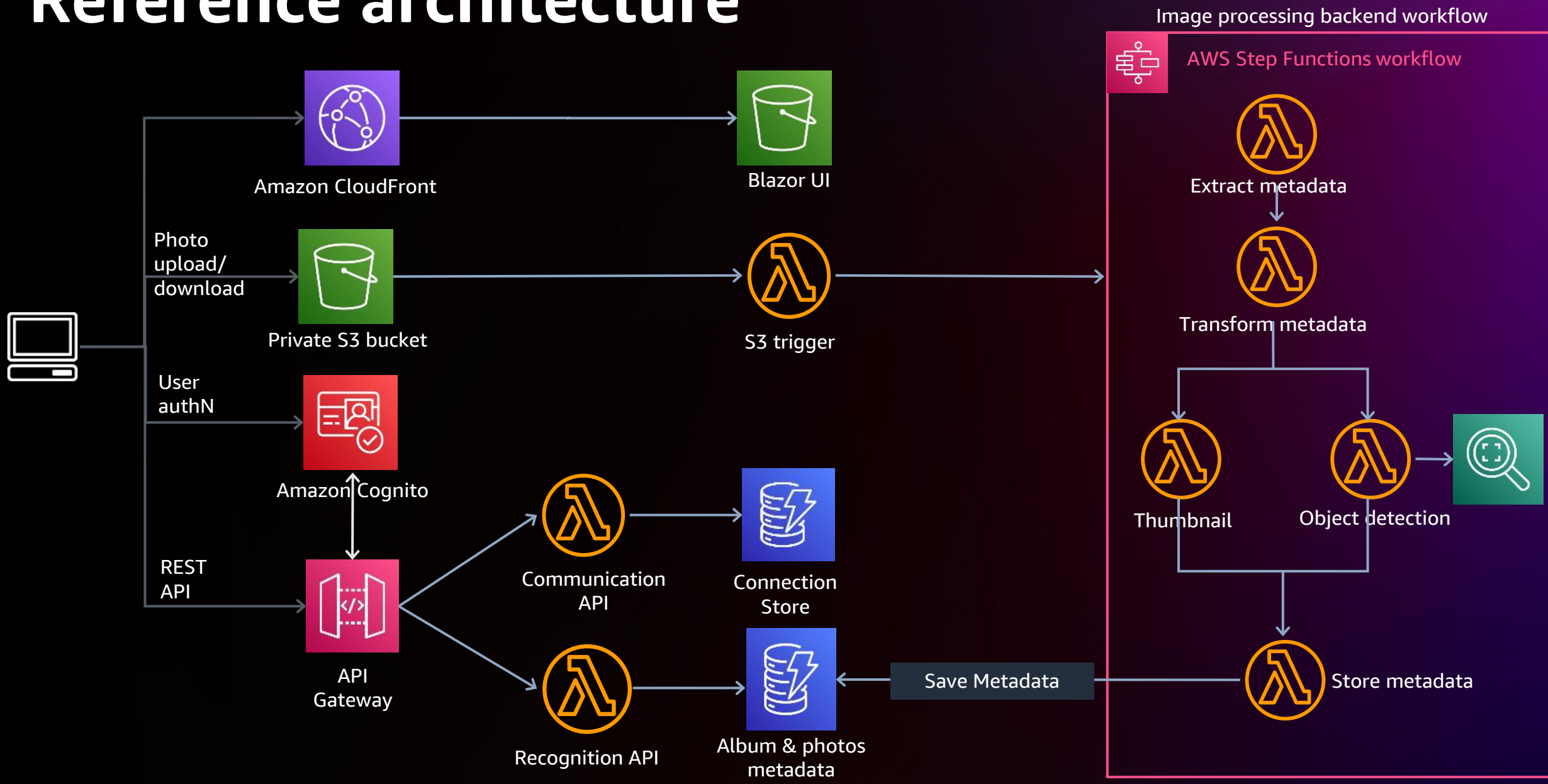
- AWS Toolkit for Visual Studio
- AWS Extensions for .NET CLI
- Mock Lambda Test Tool

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Architecture

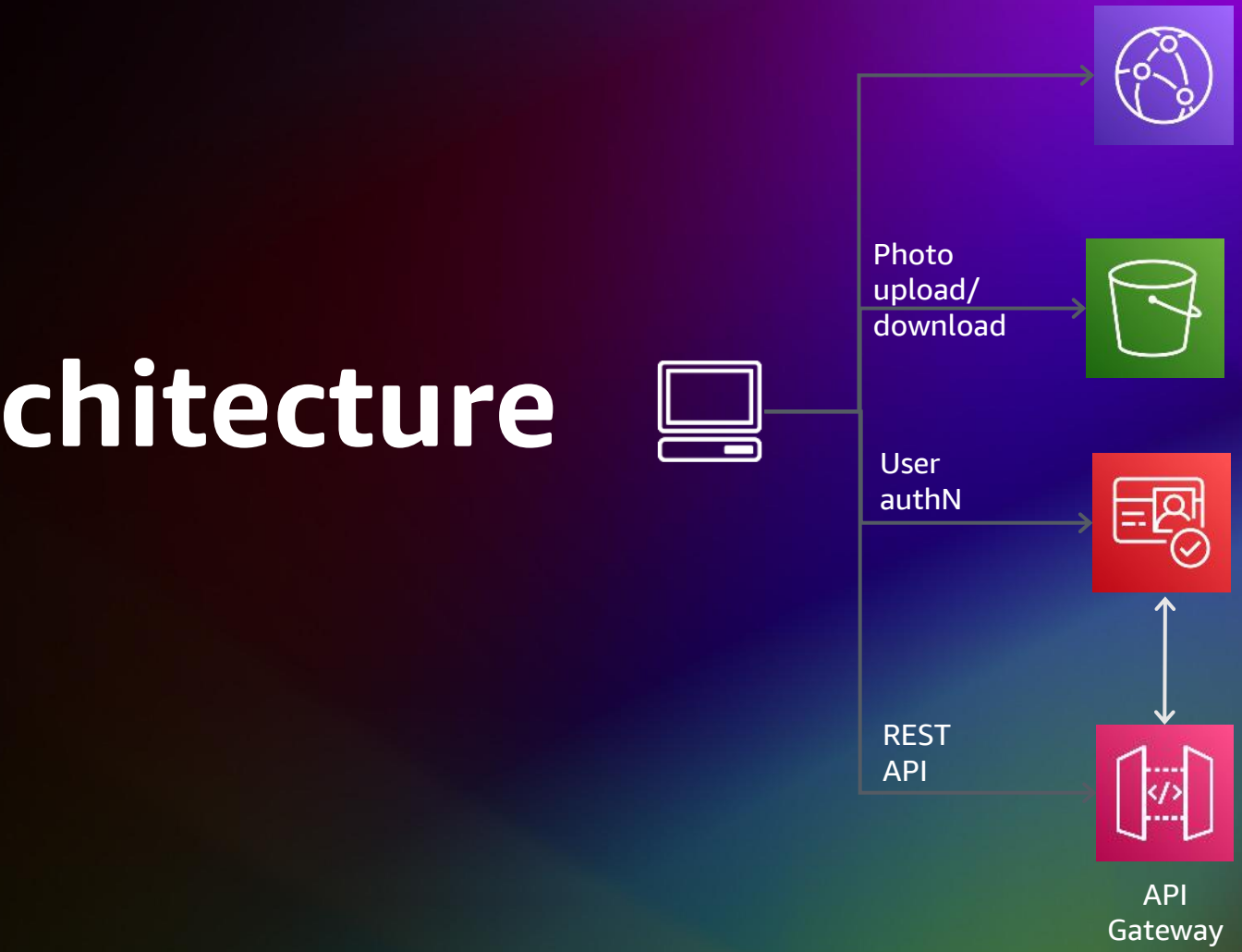
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Reference architecture



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Frontend architecture

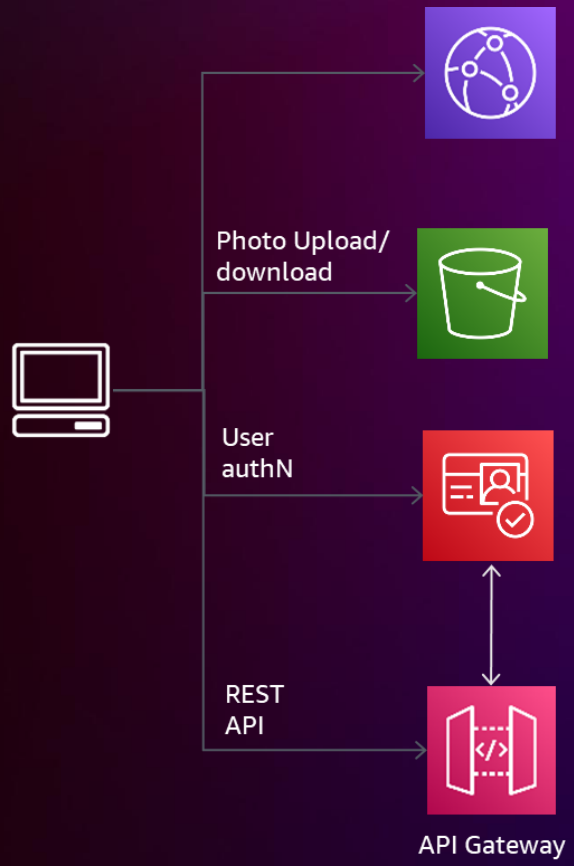


© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Application frontend architecture

CHARACTERISTICS OF A MODERN APP

- Static hosting for SPA application
- Cross-platform
- OIDC authentication

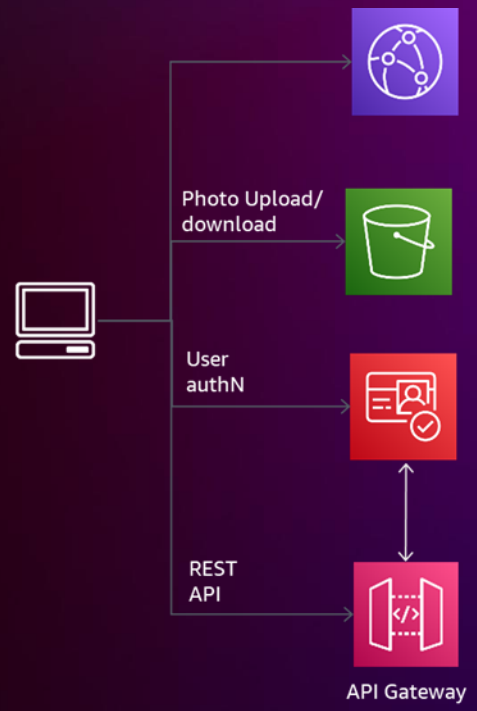


aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

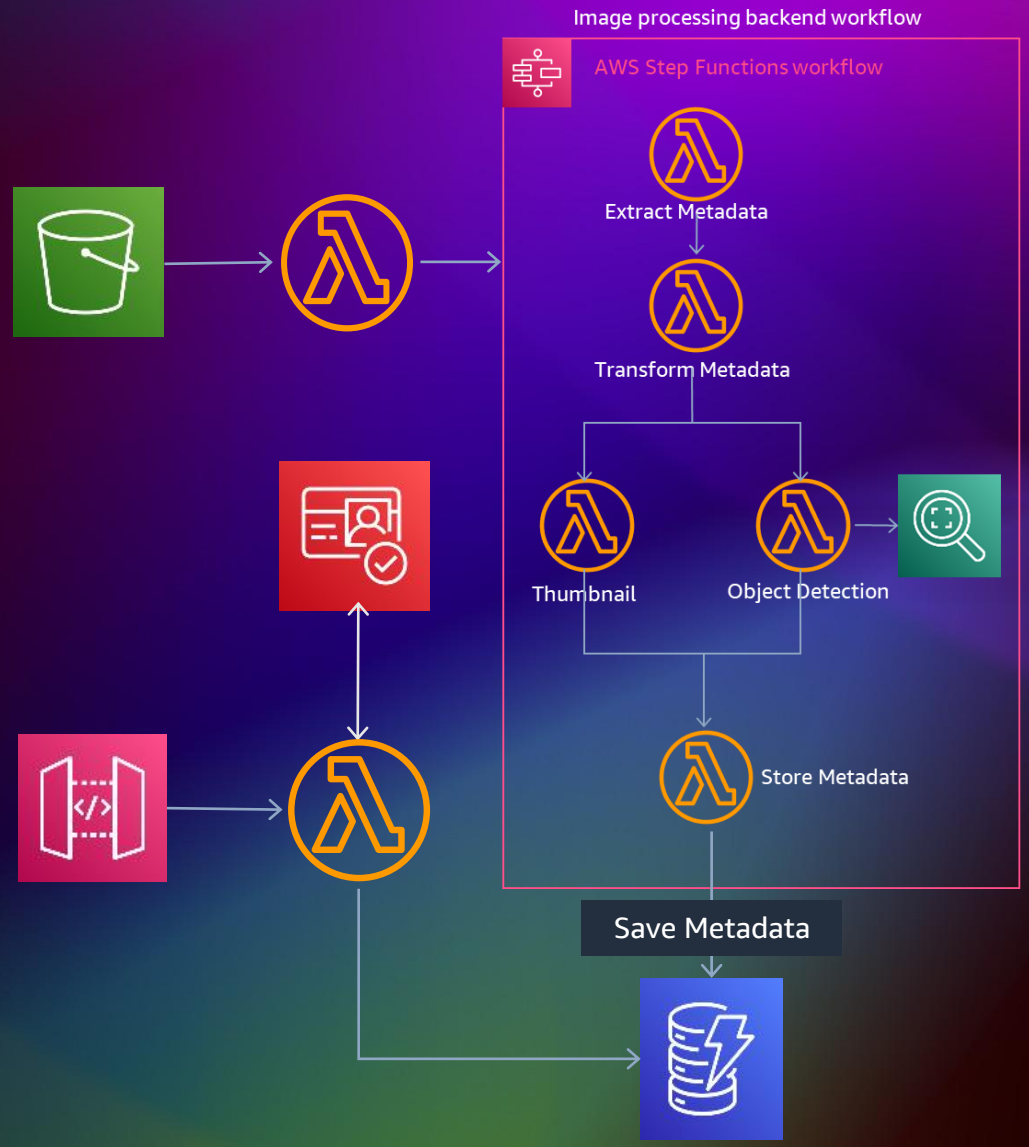
Building with .NET Blazor

- Running in WebAssembly (*Wasm*)
- Blazor Hybrid with MAUI
- OIDC Authentication with Cognito



aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Backend architecture

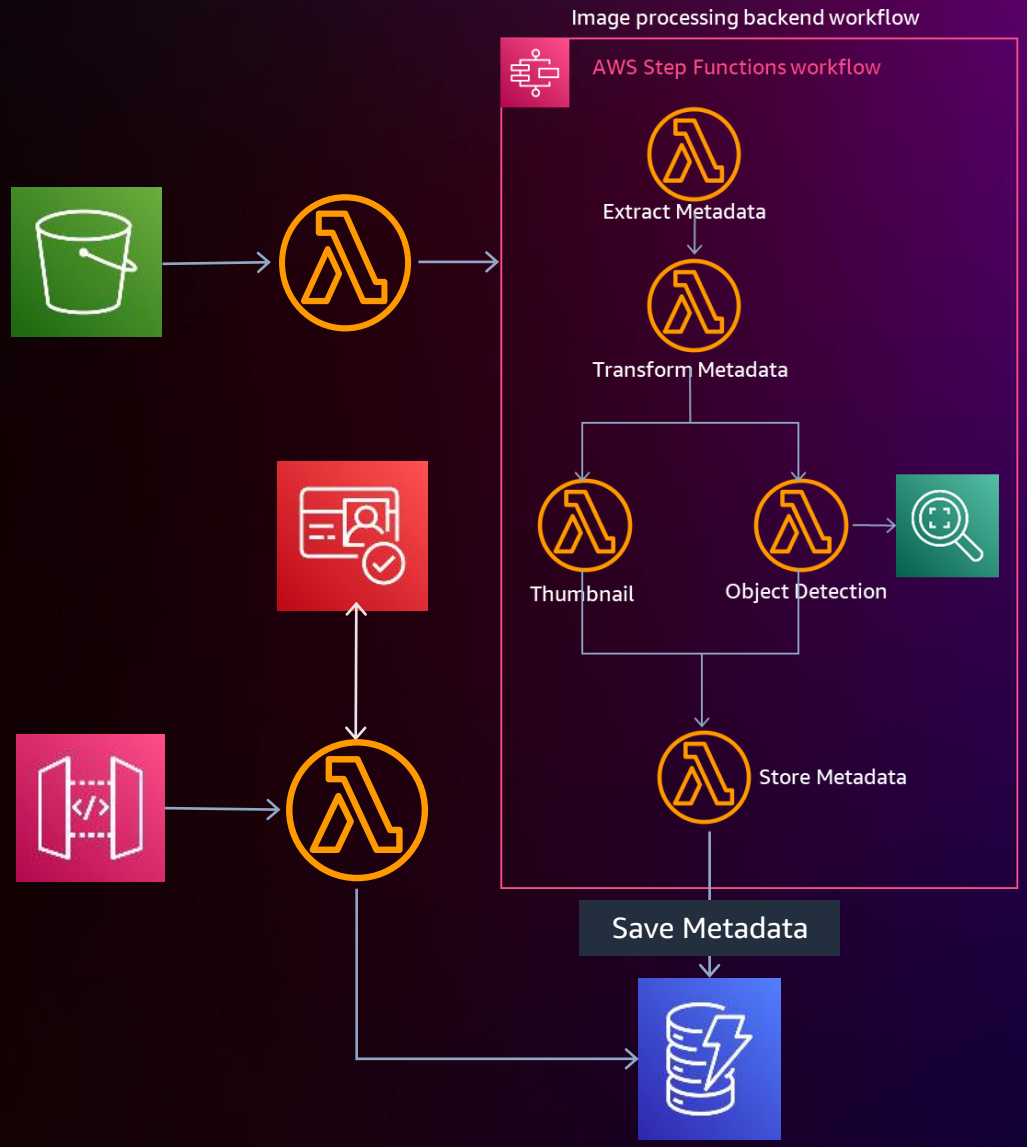


© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Backend service architecture

CHARACTERISTICS OF BACKEND SERVICE

- Event-driven and scalable
- Resilient and fault tolerant
- Secure and observable



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Building Lambda with .NET7



- Managed runtime for .NET on Lambda
- Native AOT for .NET on Lambda
- Testing .NET Lambda on local machine

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Observability matters because ...



Visibility



Real-time troubleshooting



Customer experience



Applications = $$

Operational → **Business**

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

When a bug happens how do we trace it?

Let’s debug and see how to trace the issue

Observability demo

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Get the code!!!

- Here is one way to build a .NET Modern Serverless application

- Sample code available on GitHub
  - Fork the code to use as you like
  - Deploy the working app with a single click (deploy takes about 20 minutes)
  - Tutorials and labs that leverage this .NET modern app on AWS



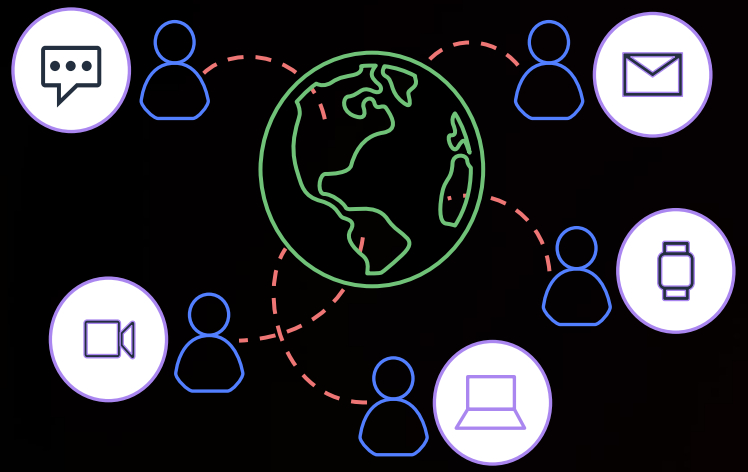
https://github.com/aws-samples/dotnet-serverless-imagerecognition

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Summary:

CHARACTERISTICS OF A MODERN APP





e-commerce



Media streaming



Social media



Online gaming



Shared economy

- Use independently scalable microservices (serverless, containers)
- Connect through APIs
- Deliver updates continuously
- Scale globally
- Are fault tolerant
- Manage state and persistence
- Have security built in
- Decoupled data strategy

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Q&A

Brian Lewis

/in/brian-lewis-it

Sanjay Gulati

/in/gulatisanjay

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Thank you!

Brian Lewis
/in/brian-lewis-it

Sanjay Gulati
/in/gulatisanjay



Please complete the session survey in the **mobile app**

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.