AWS Developer Training - .NET

Abdul Rasheed Feroz Khan
Director - CodeSizzler India | UAE
www.codesizzler.in

Hello!!

- Founder & CEO – CodeSizzler (India & UAE)
- Microsoft Most Valuable Professional – Azure
- Microsoft Certified Trainer
- AWS Super Hero
- Cloud Advocate & AI Lead - CopilotFactory
- Azure – AWS – GCP – IoT – Machine Learning – Apps & Infra – Data & AI
- Consultant for Accenture, Adobe, Boeing, Capgemini, Deloitte., Infosys, Johnson Controls, Microsoft, Wipro, etc.,

Abdul Rasheed Feroz Khan
feroz@codesizzler.in
www.linkedin.com/in/arfk

Agenda – Day 1

AWS Fundamentals & .NET Integration

Session 1: Introduction to AWS & .NET SDK (2 hrs)

- AWS Global Infrastructure (Regions, AZs)
- AWS SDK for .NET (C#) setup & configuration
- IAM (Identity & Access Management) for .NET apps
    - Roles, Policies, and .NET SDK integration

Hands-on Lab:

Deploy a .NET Core app using AWS SDK for IAM access

Session 2: Compute Services for .NET (2 hrs)

- AWS Lambda with .NET
    - Writing C# Lambda functions
    - Triggers (API Gateway, S3, DynamoDB)
- EC2 for .NET Applications
    - Deploying ASP.NET Core on EC2

Hands-on Lab:

- Build & deploy a serverless .NET API with Lambda

Session 3: Storage & Database Services (3 hrs)

- Amazon S3 – Storing & retrieving files from .NET
- DynamoDB – NoSQL with .NET SDK
    - CRUD operations in C#
    - DynamoDB Accelerator (DAX)
- RDS for SQL Server – Running managed SQL with .NET

Hands-on Lab:

- Build a .NET app with DynamoDB & S3 integration

Agenda – Day 2

Advanced Development and Security

Session 1: Serverless & Event-Driven .NET (3 hrs)

- Amazon SQS & SNS – Messaging in .NET
- EventBridge – Event-driven architectures
- Step Functions – Workflow automation with .NET

Hands-on Lab:

- Create a .NET event-driven app using SQS & Lambda

Session 2: Authentication & Security (2 hrs)

- Amazon Cognito – User auth in .NET apps
- Secrets Manager & Parameter Store – Managing .NET app secrets
- KMS & Encryption SDK – Data encryption in C#

Hands-on Lab:

- Secure a .NET Web API with Cognito

Session 3: CI/CD for .NET on AWS (3 hrs)

- AWS CodeCommit, CodeBuild, CodeDeploy, CodePipeline
  - Building & deploying .NET apps
- Infrastructure as Code (AWS CDK in C#)

Hands-on Lab:

- Set up a CI/CD pipeline for a .NET Core app

Agenda – Day 3

Monitoring, Debugging and Final Review

Session 1: Monitoring & Logging (3 hrs)

- CloudWatch Logs & Metrics – .NET app monitoring
- AWS X-Ray – Distributed tracing in .NET

Hands-on Lab:

- Debug a .NET Lambda function with X-Ray

Session 2: Performance Optimization & Cost (2 hrs)

- Caching with ElastiCache (Redis) for .NET
- Optimizing Lambda & API Gateway for .NET
- AWS Cost Explorer & Budgets

Session 3: Mock Exam & Q&A (3 hrs)

- Practice Test Review (AWS Sample Questions)
- Troubleshooting Common .NET on AWS Issues
- Final Tips for Exam Success

Pre-test

https://www.surveymonkey.com/r/5KMCZ3N

ode $izzler

AWS Fundamentals & .NET Integration

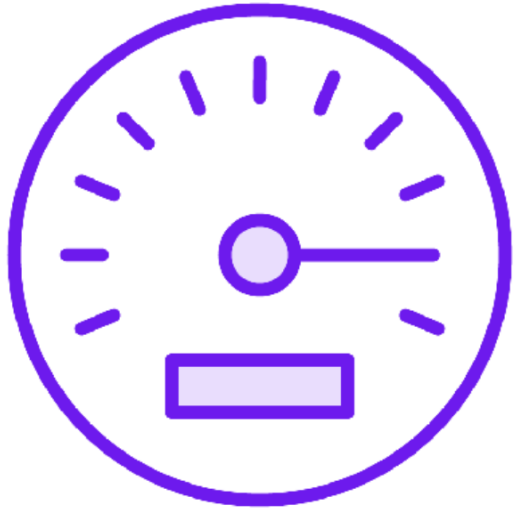
Introduction to AWS & .NET SDK

AWS Global Infrastructure – Cloud Computing?

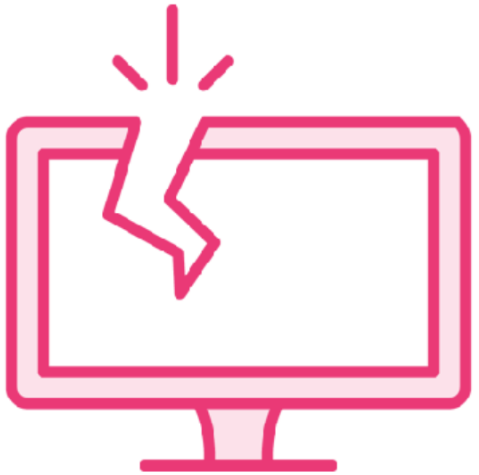
"Cloud computing is essentially the on-demand delivery of IT resources over the internet with pay-as-you-go pricing."

Three Major Challenges

Scalability

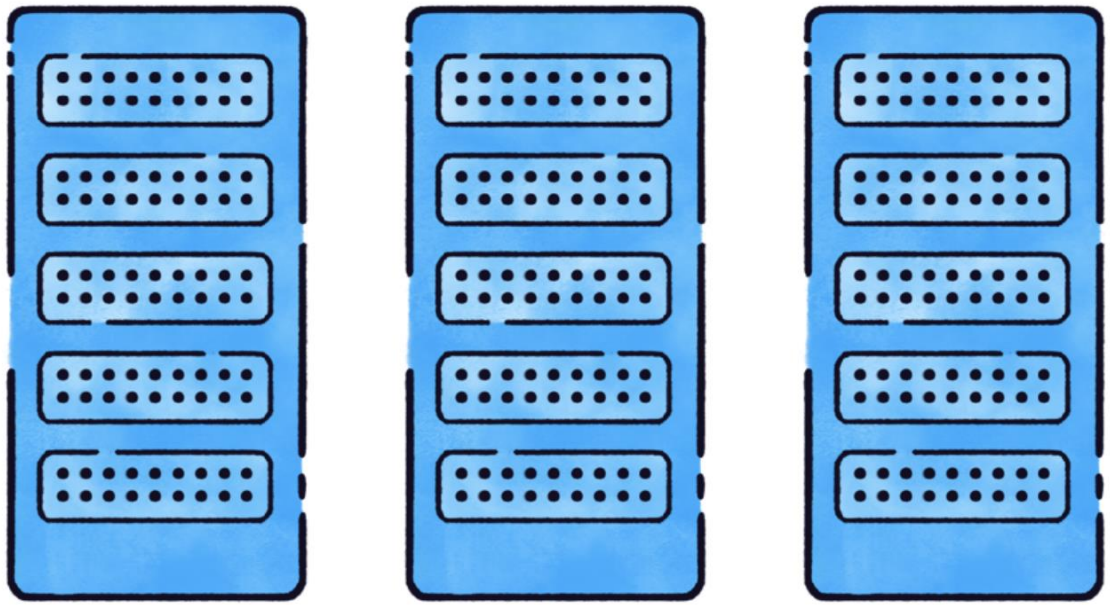


Performance

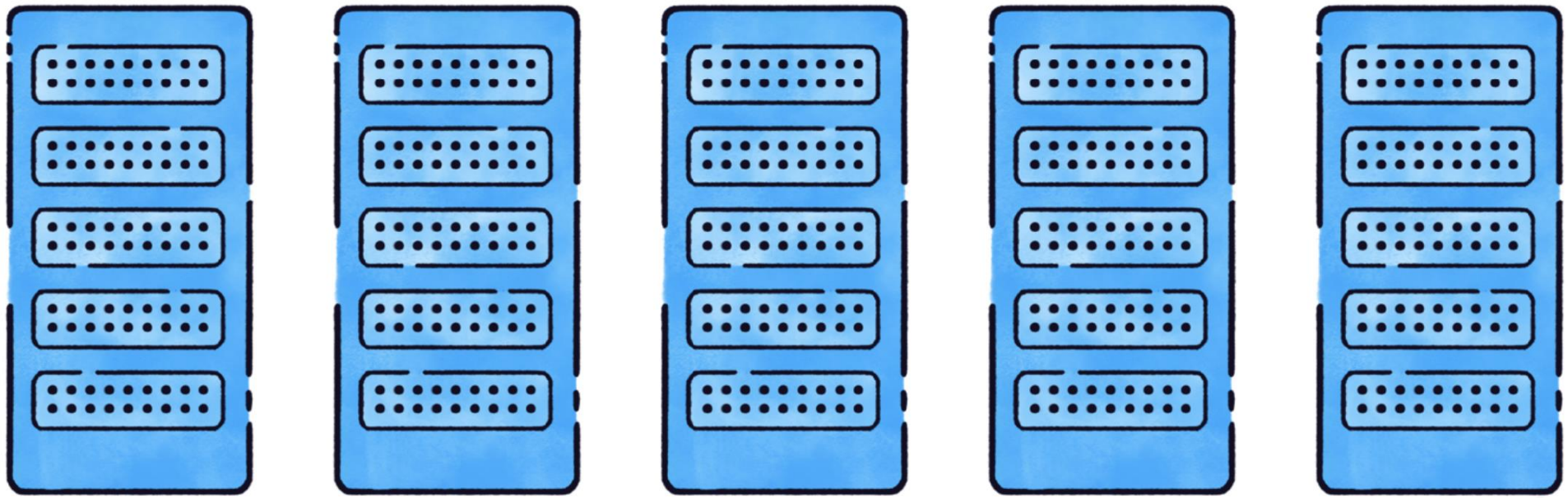


Fault Tolerance

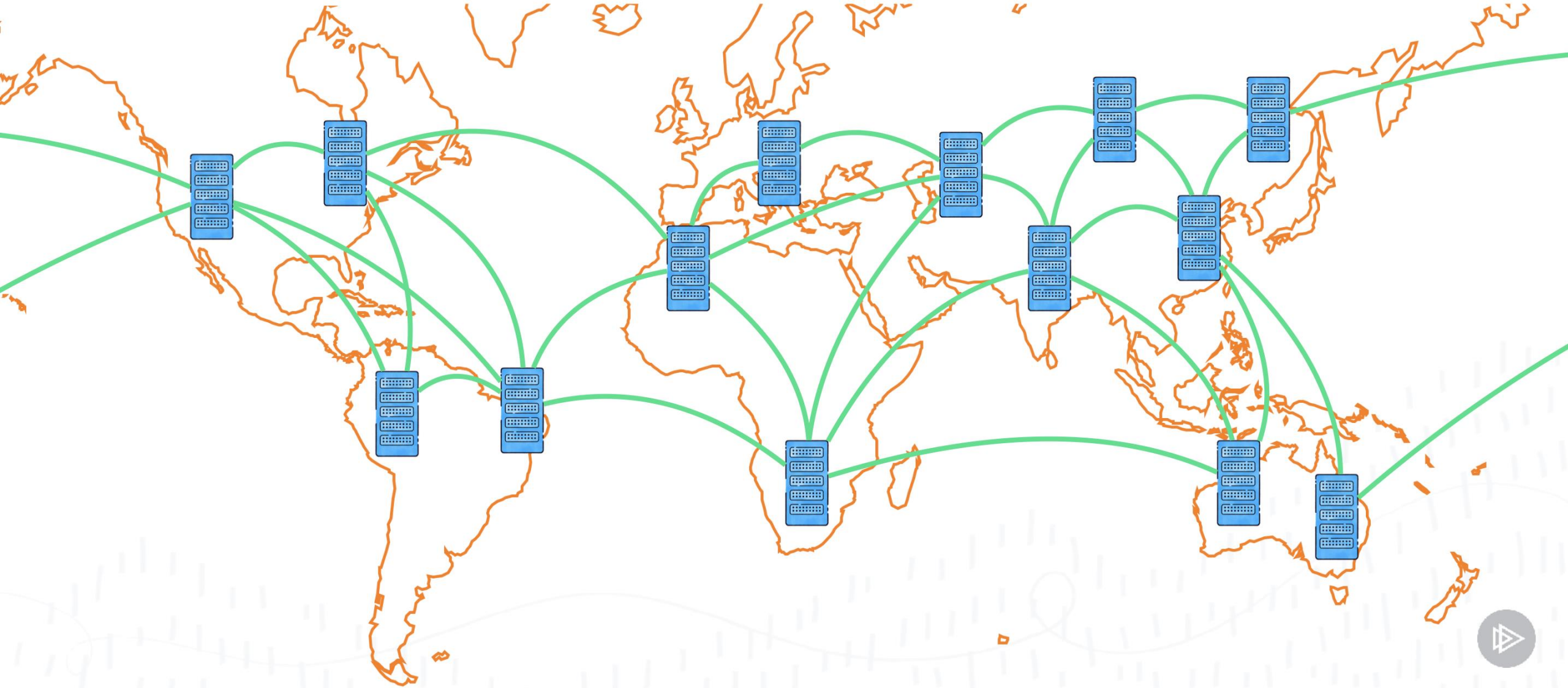
Scalability



Scalability



The Cloud



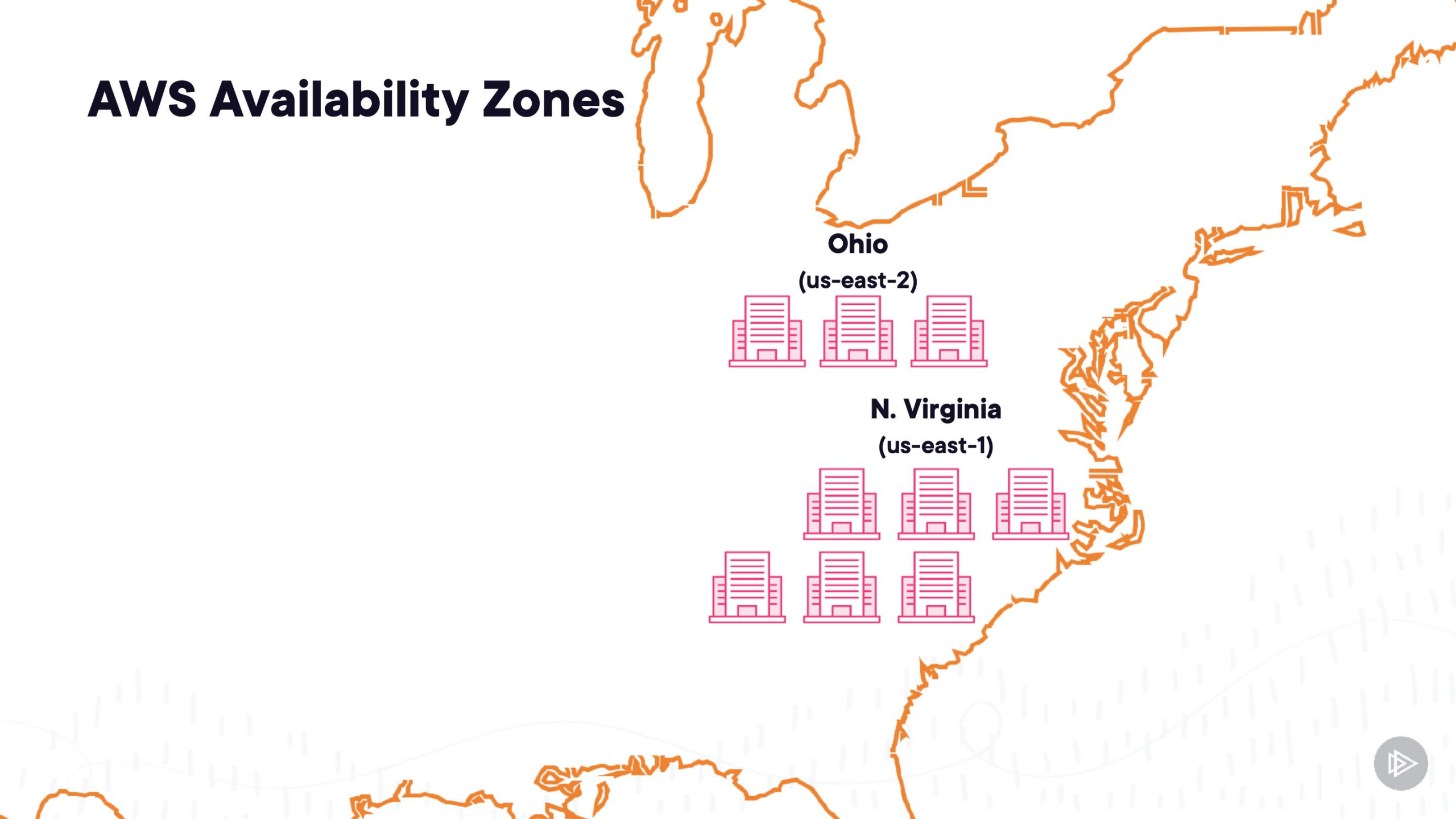
Regions and Availability Zones

AWS Regions

AWS Regions

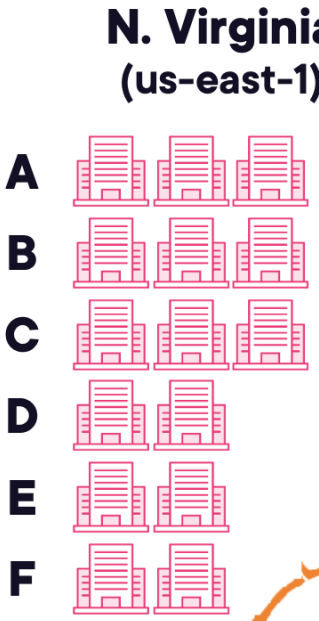

Oregon
(us-west-2)
N. California
(us-west-1)
Ohio
(us-east-2)
N. Virginia
(us-east-1)

AWS Availability Zones

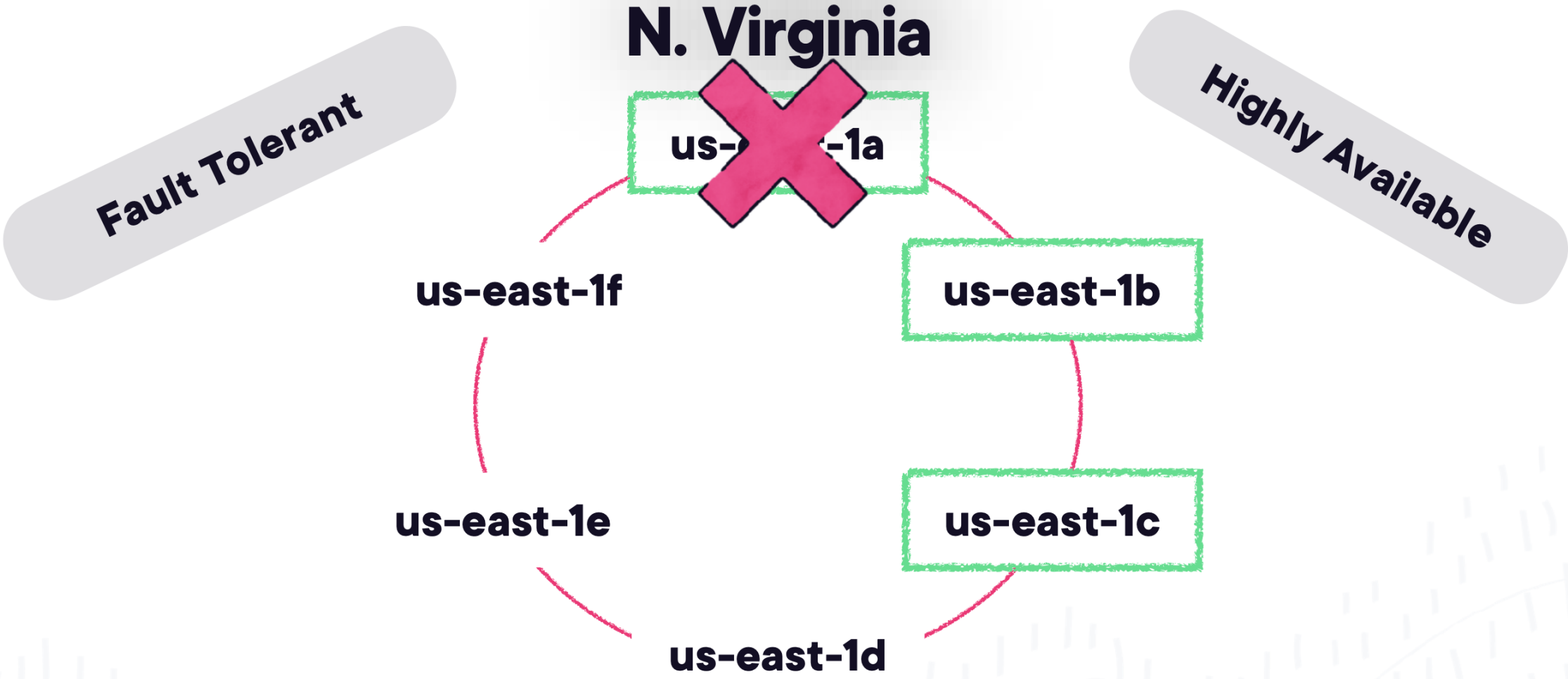


AWS Availability Zones

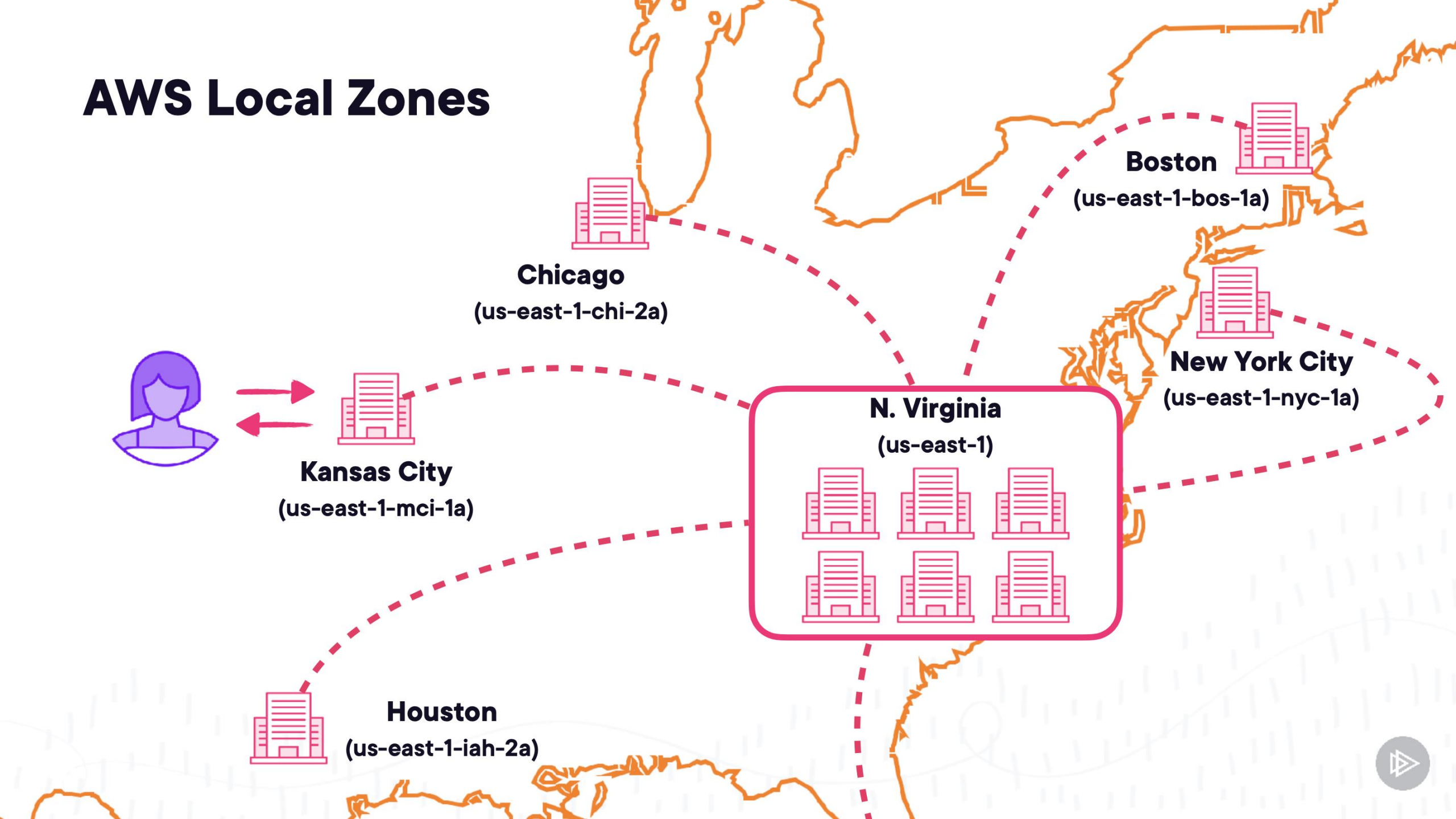
One or more discrete data centers with redundant power, networking, and connectivity in an AWS Region.



AWS Availability Zones



AWS Local Zones



AWS Local Zones

AWS Local Zones have limited capabilities, but allow you to achieve low latency by deploying applications closer to your end users. There are over 30 Local Zones in major metropolitan areas around the world.

AWS Local Zones

AWS Local Zones can also help you adhere to **data residency** restrictions, by storing and processing data in specific geographical regions.

AWS Edge Locations

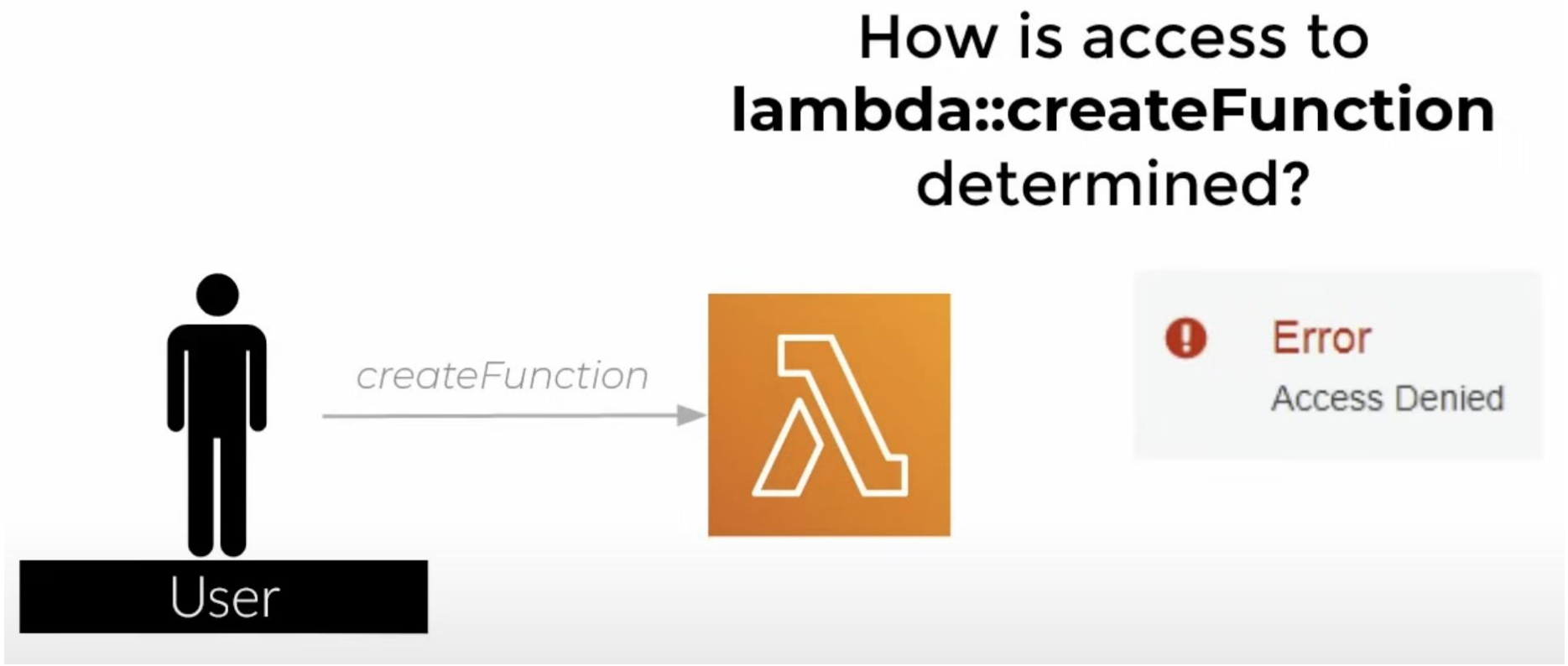
IAM

Core Concepts

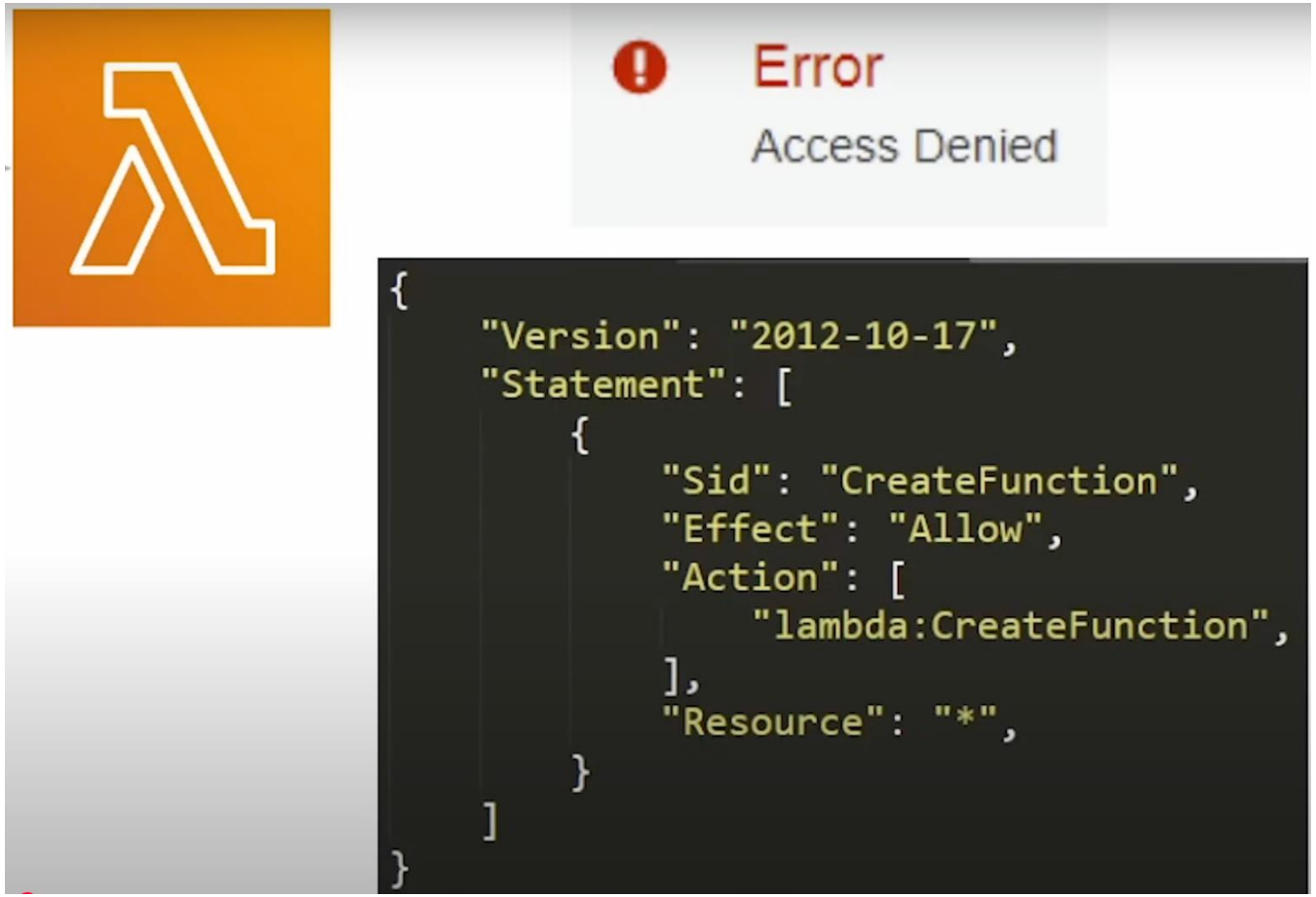
Identity and Access Management (IAM)

- Core AWS Service that helps you control access to **Resources**
- **Resources** are the entities you create in AWS i.e. S3 Bucket or Object
- Users attempt to perform **Actions** on resources, i.e. S3::CreateBucket
- Authorization to perform an **Action** depends on a **Policy**

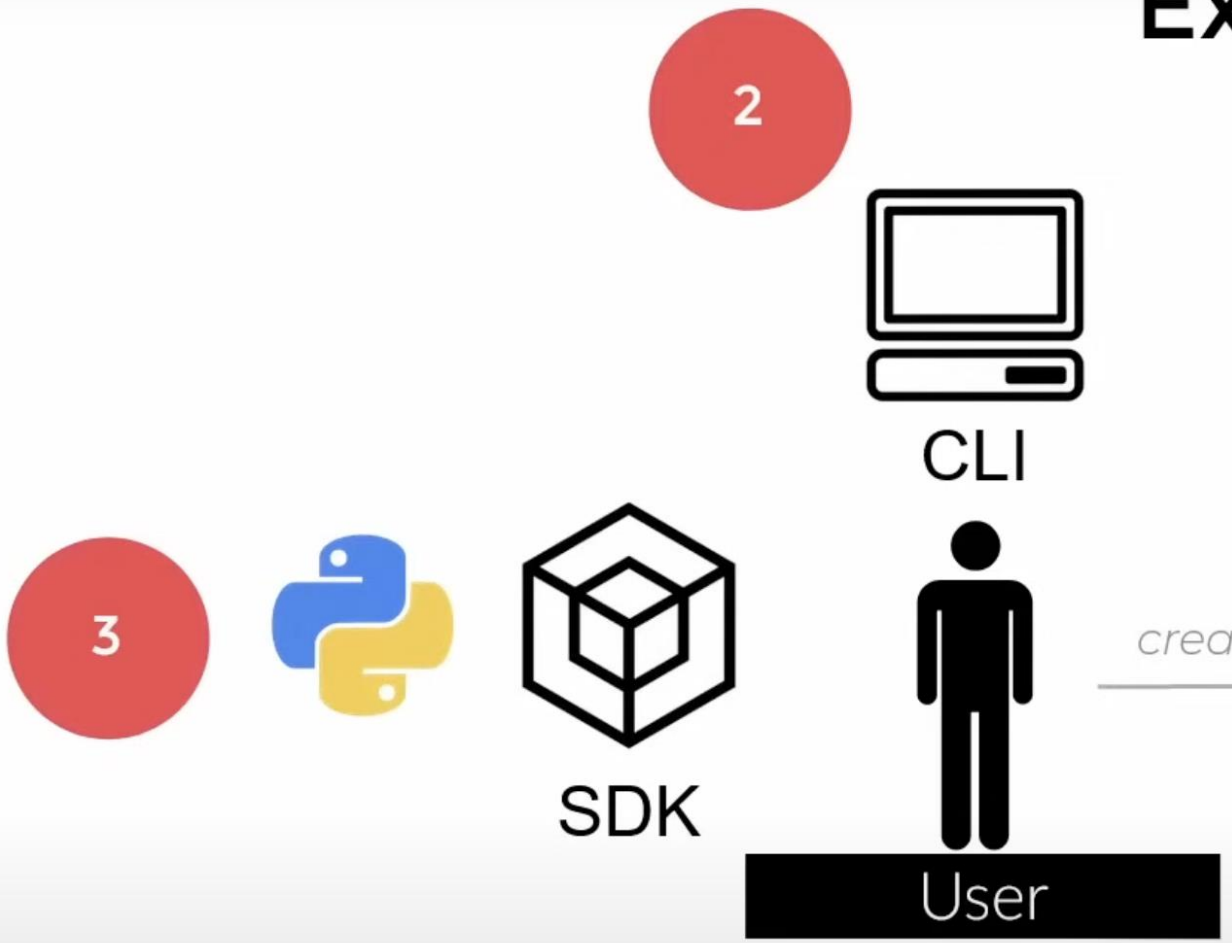
Example


How is access to
lambda::createFunction
determined?
createFunction
User
Error
Access Denied

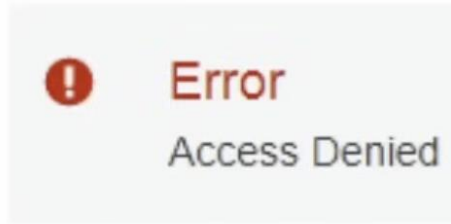
Example (cont.)



Example



How is access to **lambda::createFunction** determined?



```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "CreateFunctio
            "Effect": "Allow",
            "Action": [
                "lambda:CreateFu
            ],
            "Resource": "*",
        }
    ]
}
```

Detour: Access Key & Secret Access Key

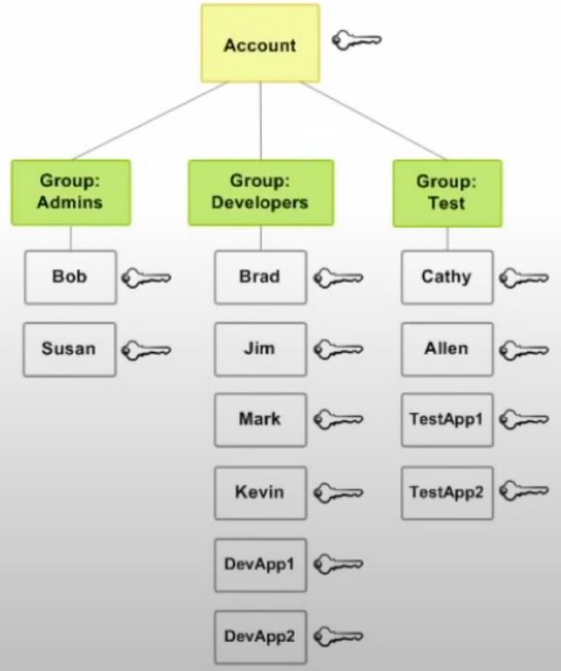
- **Secret** codes that you use to interact with AWS

```
lambdaClient = boto3.client('lambda',
                            aws_access_key_id=settings.AWS_SERVER_PUBLIC_KEY,
                            aws_secret_access_key=settings.AWS_SERVER_SECRET_KEY,
                            region_name=REGION_NAME
                            )
lambdaClient.createFunction( ... )
```

```
PS C:\Users\Dan> aws configure
AWS Access Key ID [****************WM6C]:
AWS Secret Access Key [****************gEmj]:
Default region name [us-east-1]:
Default output format [None]:
```

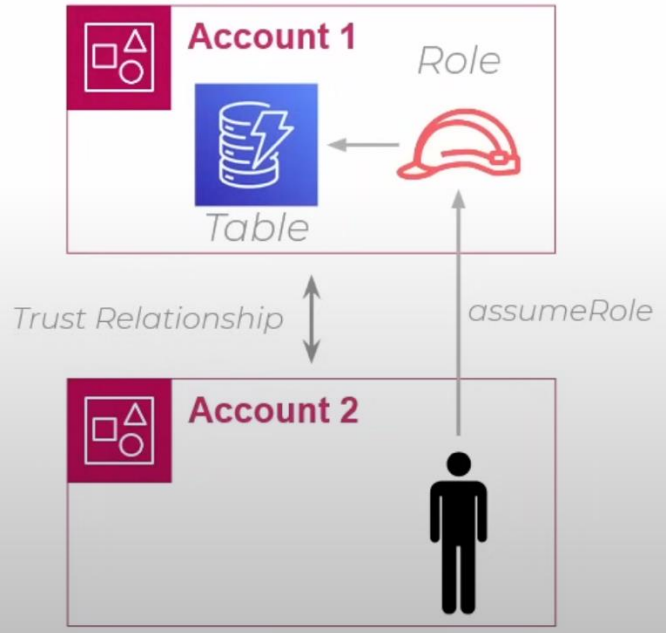
Other Important Concepts

Groups



Roles

Trust Relationships



AWS SDK for .NET (C#) setup & configuration

AWS SDK for .NET - https://aws.amazon.com/sdk-for-net/

Download AWS Toolkit from VS Code Marketplace

File Edit Selection View Go ··· Search

EXPLORER ···

OPEN EDITORS

Extension: AWS Toolkit

NO FOLDER OPENED

You have not yet opened a folder.

Open Folder

Opening a folder will close all currently open editors. To keep them open, add a folder instead.

You can clone a repository locally.

Clone Repository

To learn more about how to use Git and source control in VS Code read our docs.

You can open a folder containing a .NET project or solution, or create a new .NET project.

Create .NET Project

Extension: AWS Toolkit

AWS Toolkit

Amazon Web Services amazon.com | 3,246,409 | (71)

Including CodeCatalyst, Infrastructure Composer, and support for Lambda, S3, Cl...

Installing Auto Update

DETAILS FEATURES CHANGELOG

Demo & Usecase – Deploy a .NET Core app using AWS SDK for IAM access

Lab time – 30 minutes

Questions

ode $izzler

AWS Fundamentals & .NET Integration

Compute Services for .NET

What will we talk?

- AWS Lambda with .NET
  - Writing C# Lambda functions
  - Triggers (API Gateway, S3, DynamoDB)
- EC2 for .NET Applications
  - Deploying ASP.NET Core on EC2

Hands-on Lab:

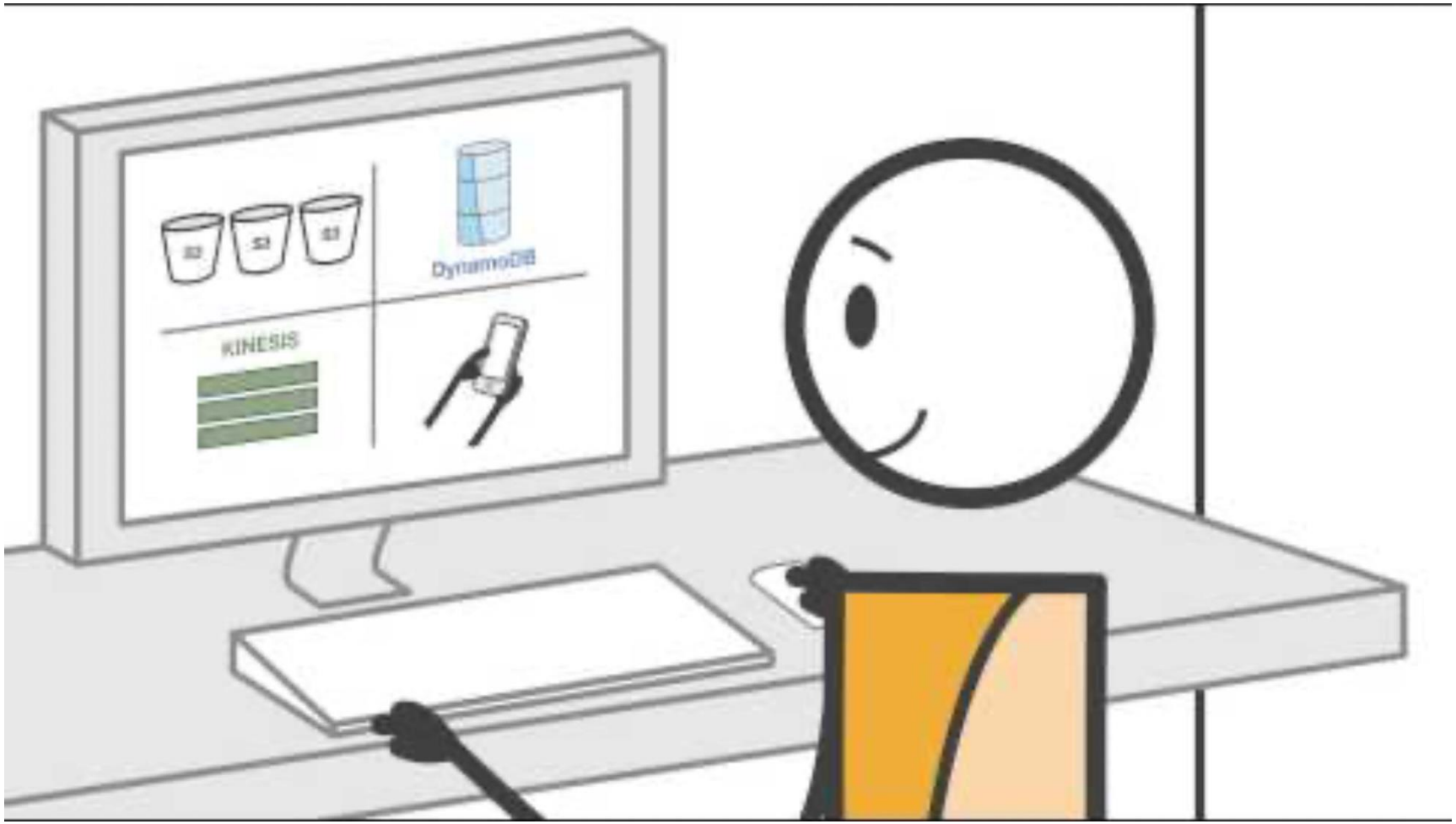
- Build & deploy a serverless .NET API with Lambda

Serverless?

Pricing

Product	Description	Free Tier Offer Details	Product Pricing
AWS Lambda Serverless Computing	AWS Lambda is a compute service that runs your code in response to events and automatically manages the compute resources.	This always free service is on the **Free and Paid pla**n. Use your credits to evaluate beyond these monthly limits: 1,000,000 free requests per month Up to 400,000 GB-seconds or 3.2 million seconds of compute time per month	AWS Lambda Pricing

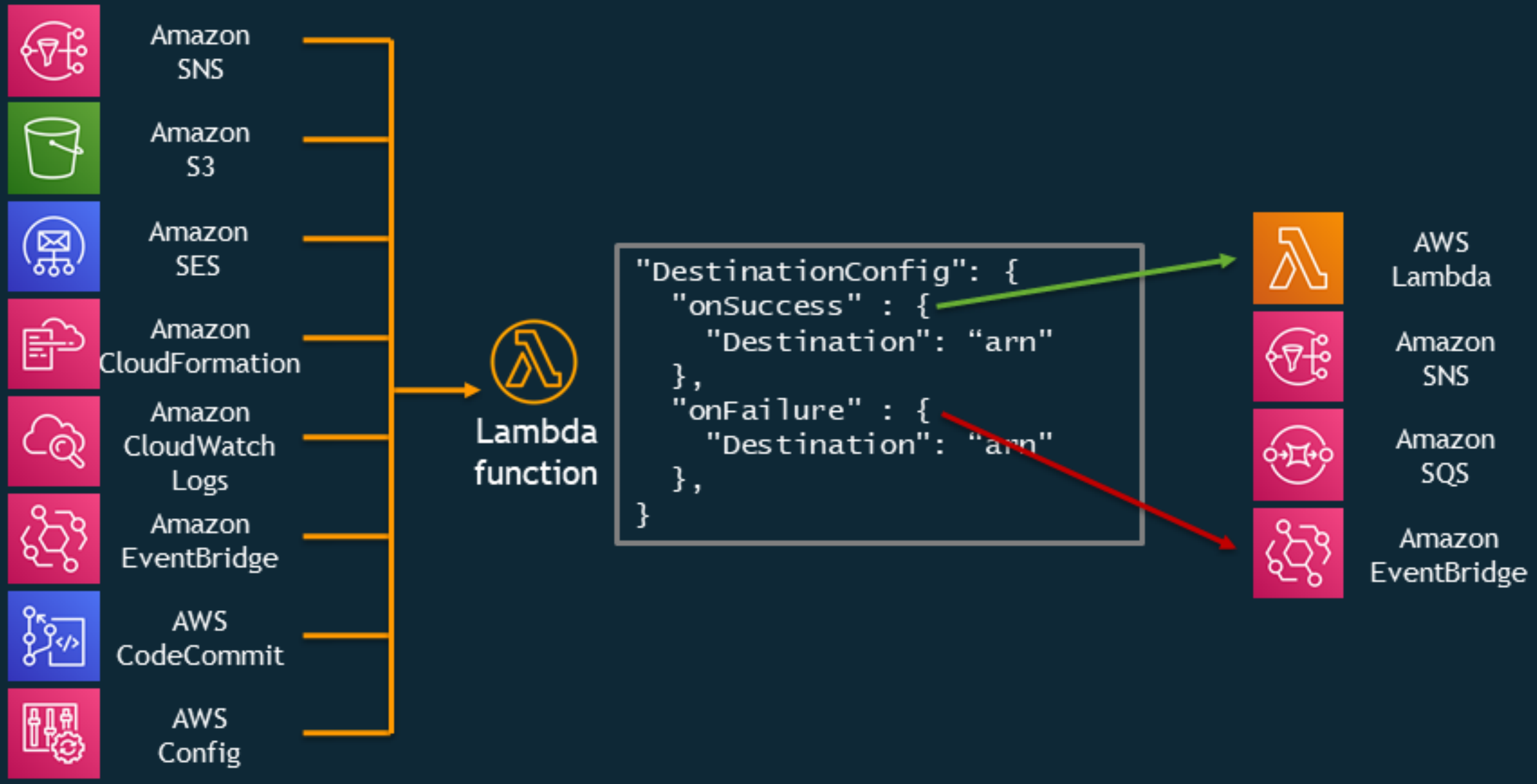
AWS Lambda



When to use?
When not to use?

Triggers

Asynchronous Function Execution Result

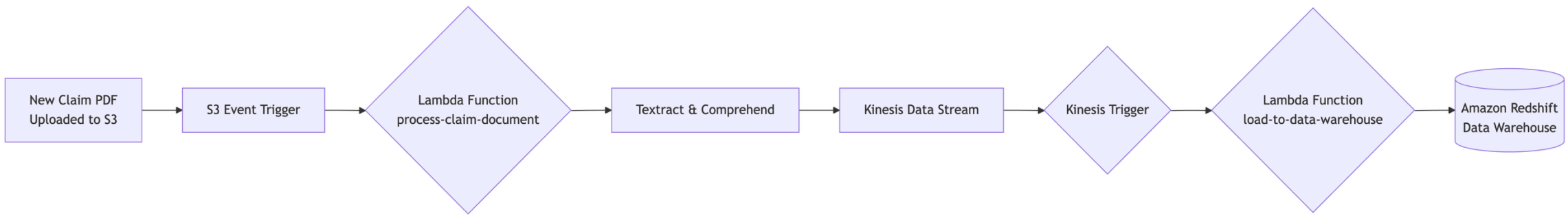


Use case

An insurance company receives thousands of claims documents (PDFs, images) daily from various channels (email, web portal, agent uploads). They need to:

1. Immediately process each new document when it arrives.
2. Extract text and data from the documents.
3. Analyze the content for fraud detection and triage urgency.
4. Store the processed data in their data warehouse for deeper analytics.

How AWS Lambda Triggers Solve This??


New Claim PDF Uploaded to S3
S3 Event Trigger
Lambda Function process-claim-document
Textract & Comprehend
Kinesis Data Stream
Kinesis Trigger
Lambda Function load-to-data-warehouse
Amazon Redshift Data Warehouse

Demo – Quick tour on AWS Lambda Function with .NET

EC2 for .NET applications

Amazon EC2

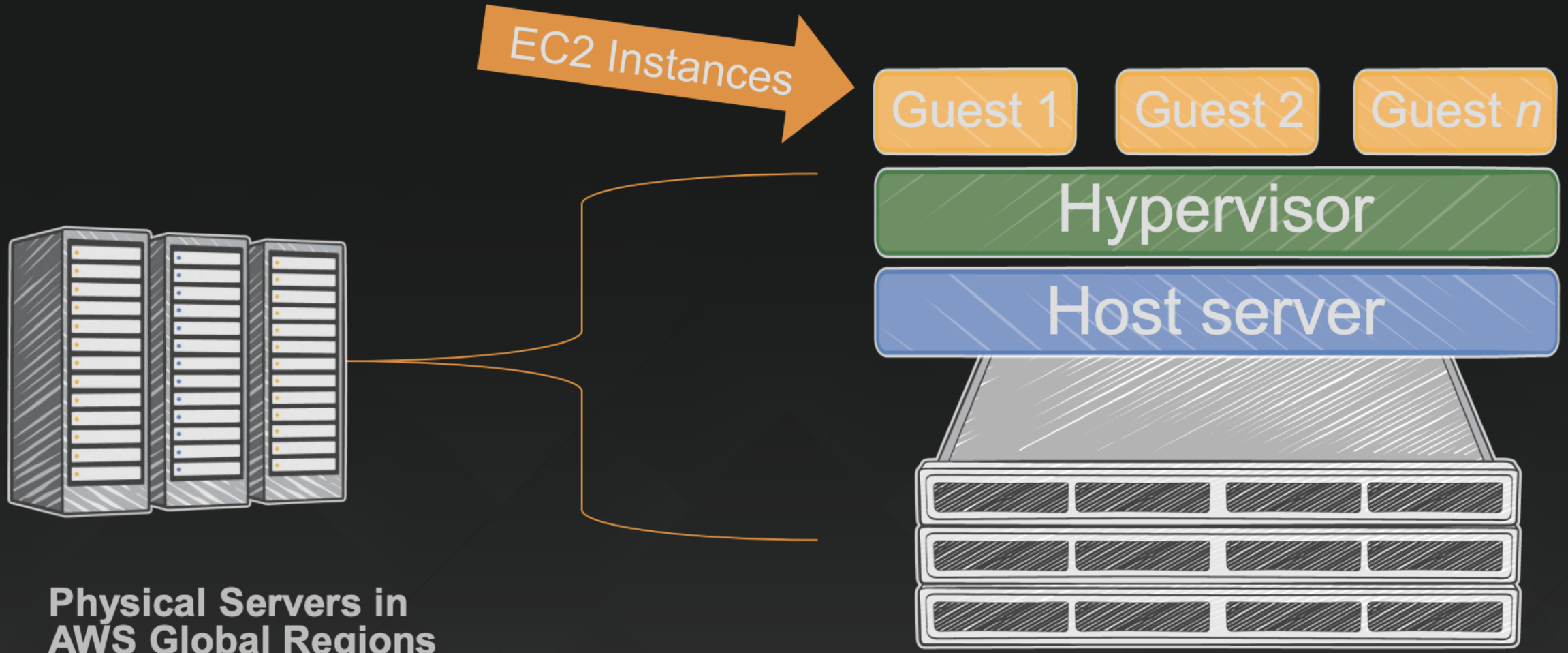
Virtual machines

Purchase options

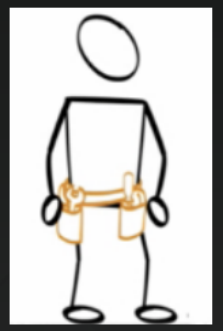
Networking

User experience

Amazon Elastic Compute Cloud (EC2) - Elastic virtual servers in the cloud



EC2 instances: Families and Generations



General-purpose:	M1, M3 , M4, T2
Compute-optimized:	C1, CC2, C3, C4
Memory-optimized:	M2, CR1, R3
Dense-storage:	HS1, D2
I/O-optimized:	HI1, I2
GPU:	CG1, G2
Micro:	T1, T2

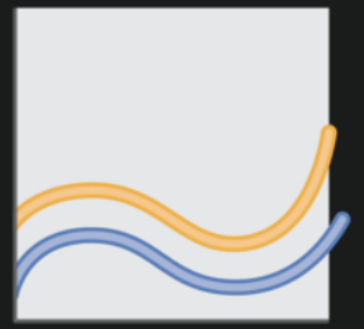
Why do customers use Amazon EC2?

amazon web services

Why Do Customers Use Amazon EC2?

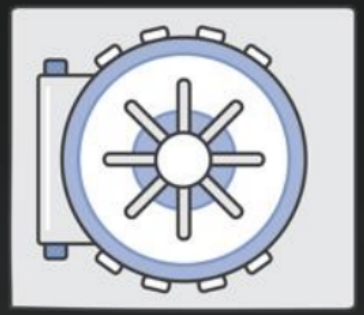
Fast Deployments
Access computing infrastructure in minutes

Low Cost
Pay-as-you-go pricing



Elastic
Easily add or remove capacity

Globally Accessible
Easily support customers around the world



Secure
A collection of tools to protect data and privacy

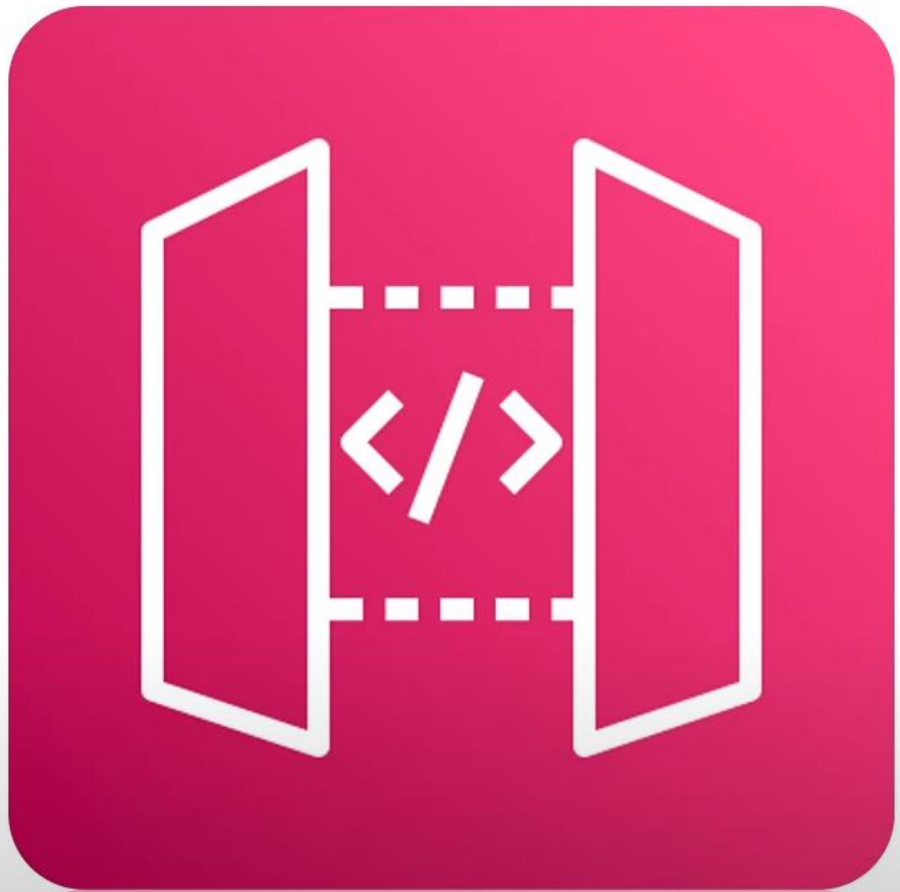
Scalable
Access to effectively limitless capacity

amazon web services

Demo – EC2 with .NET

Lab time – 30 minutes

Questions



API Gateway

OVERVIEW

Key Ideas

Managed Service for **HTTP, REST**, and **WebSocket** Endpoints

HTTP & **REST** Endpoints support Request/Response Model

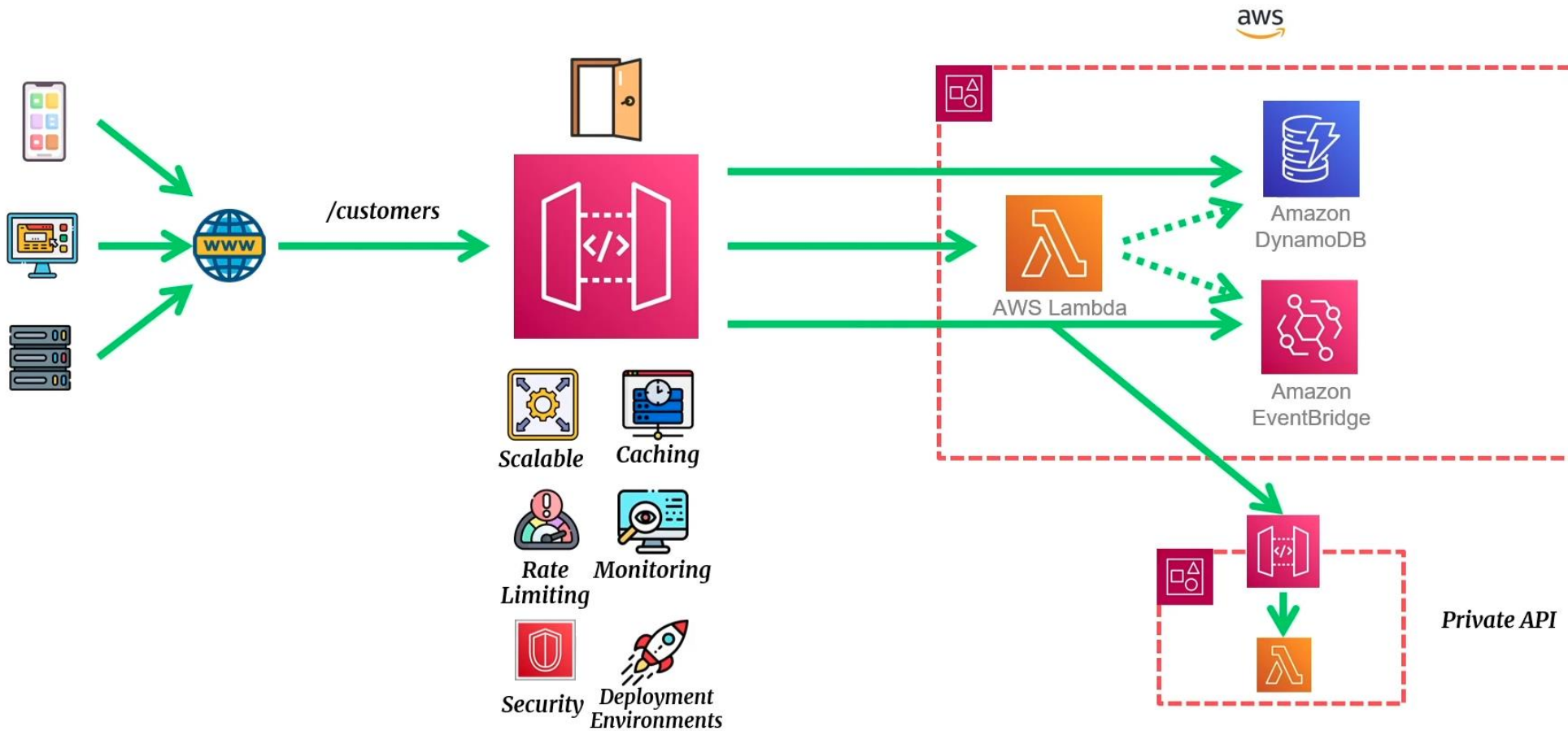
WebSockets Endpoints support bi-directional messages & broadcasting

Supported Features are different depending on HTTP, REST, or WebSockets

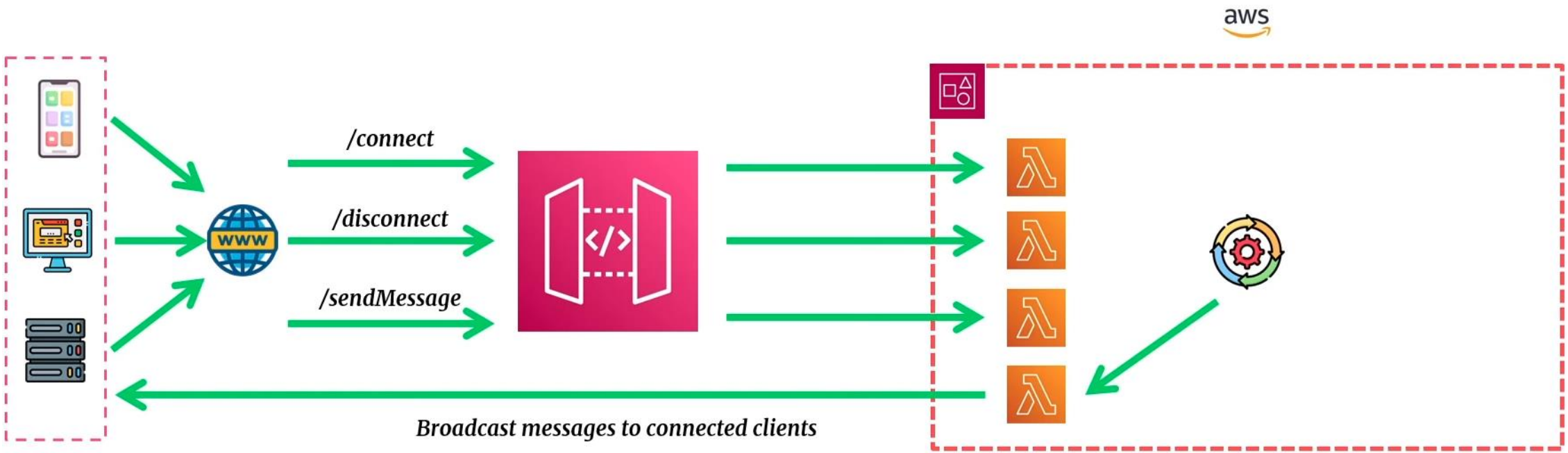
HTTP, REST, & WebSockets

HTTP	REST	WebSockets
Basic feature set allowing you to build low latency cost effective REST APIs	Enhanced REST API features adding extra functionality at a higher cost and higher latency	Enables real time bi-directional communication between client and server

HTTP / REST Example

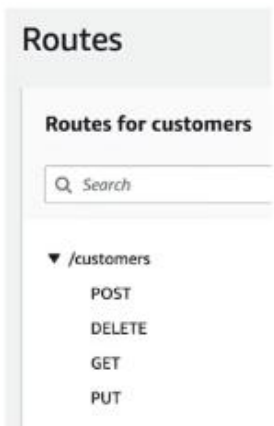


WebSocket Example

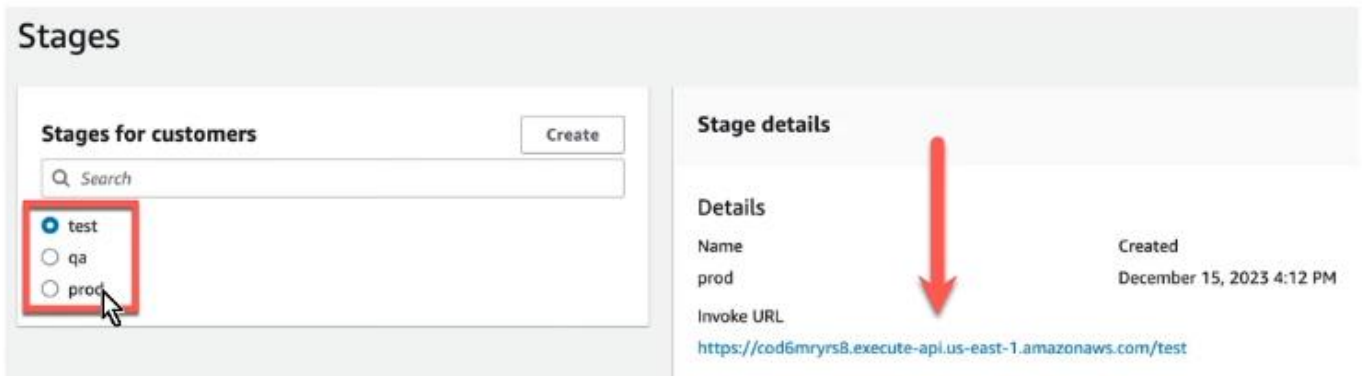


Concepts

Routes
(Resources)



Stages

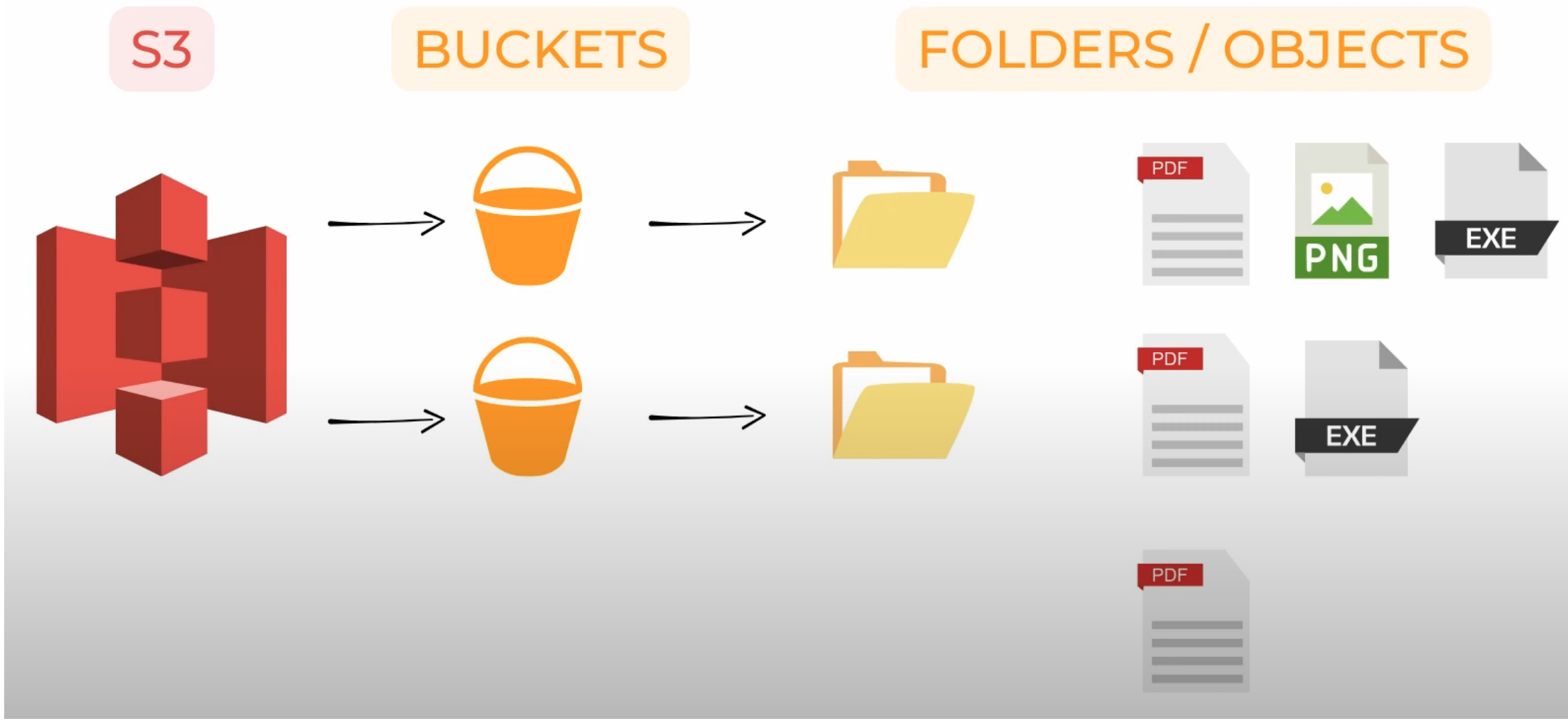


Questions?

ode $izzler

AWS Fundamentals & .NET Integration

Storage and Database Services


S3
BUCKETS
FOLDERS / OBJECTS
PDF
PNG
EXE
PDF
EXE
PDF

Pre-requisites

VISUAL STUDIO IDE

AWS CLI PROFILE CONFIGURED

.NET INSTALLED

AWS ACCOUNT

Amazon S3

Fully Managed Object Storage: Designed to store and retrieve any amount of data from anywhere on the web.

Virtually Unlimited Scalability: No need to worry about storage capacity. It scales automatically.

High Durability & Availability:

Durability: Data is automatically distributed across a minimum of 3 Availability Zones (AZs). It's designed to survive the loss of an entire AZ.

99.99% Availability: Designed

Use Cases: Backup & restore, data archiving, big data analytics, static website hosting, mobile applications, and more.

Key Concepts: Buckets and Objects

Amazon S3 Bucket

A **container** for storing objects.

Similar to a folder or a directory.

Bucket names must be GLOBALLY UNIQUE across all of AWS.

You can control access (who can create, delete, and list objects) at the bucket level.

Amazon S3 Object

The **basic entity** stored in a bucket.

An object consists of:

- **Key** (The object's name, e.g., projects/my-project/image.jpg)
- **Data** (The actual file content)
- **Version ID** (For versioning)
- **Metadata** (Key-value pairs about the data)

How Do You Interact with S3?

AWS Management Console: Easy-to-use web interface.

AWS Command Line Interface (CLI): Control AWS services from your terminal.

Example: aws s3 cp my-file.txt s3://my-bucket/

AWS SDKs: Use code in your preferred language (Python, Java, JavaScript, etc.) to interact with S3.

REST APIs: Make direct HTTP requests to S3 endpoints.

Core Storage Classes (Tiering)

Storage Class	Designed For	Minimum Storage Duration	Retrieval Time	Cost
S3 Standard	Frequently accessed data	None	Milliseconds	Highest
S3 Intelligent-Tiering	Unknown/Changing access	None	Milliseconds	Auto-optimizes
S3 Standard-IA	Infrequent, rapid access	30 days	Milliseconds	Lower
S3 One Zone-IA	Infrequent, non-critical data	30 days	Milliseconds	Very Low
S3 Glacier	Long-term archive	90 days	Minutes to Hours	Very Low
S3 Glacier Deep Archive	Long-term legal backup	180 days	12 Hours	Lowest

Essential Features - Versioning

- **What it is:** Keeps multiple variants of an object in the same bucket.
- **Why use it?**
  - **Accidental Deletion:** You can restore a previous version.
  - **Accidental Overwrites:** You can roll back to an older state.
- **How it works:** When you enable versioning for a bucket, S3 automatically assigns a unique version ID to each object stored.

Essential Features - Security & Encryption

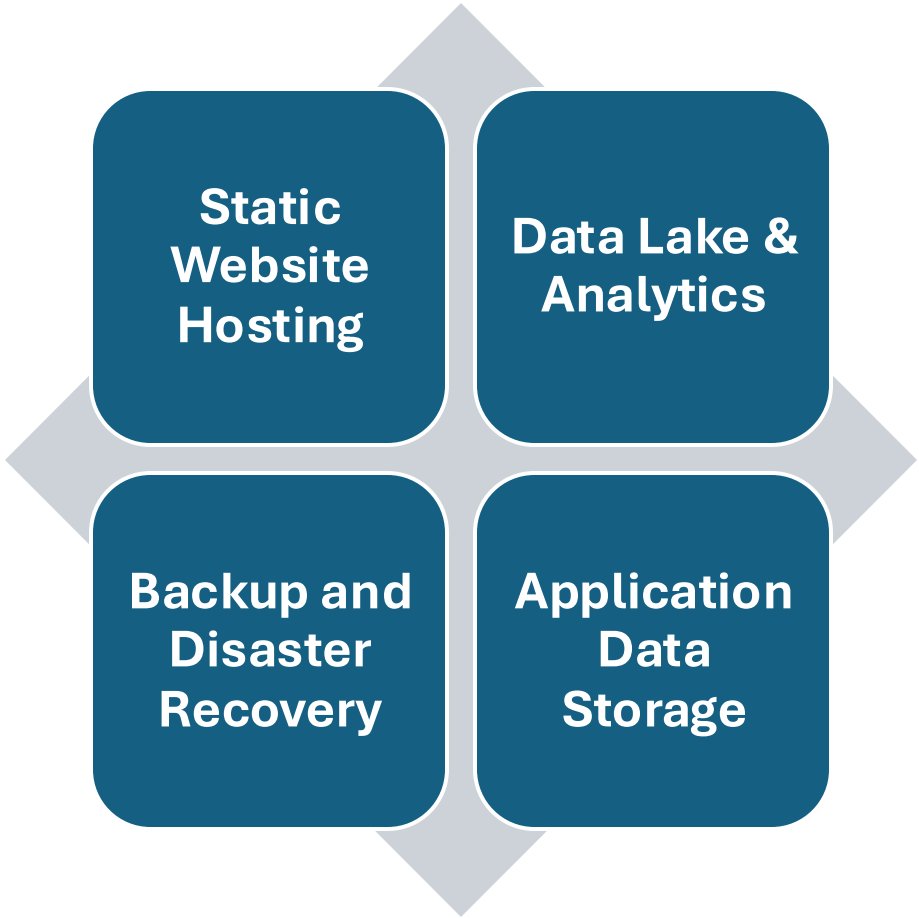
Access Management:

- **IAM Policies**
- **Bucket Policies**
- **ACLs (Legacy)**

Encryption:

- **Encryption in Transit**
- **Encryption at Rest:**
  - **SSE-S3:** S3 Managed Keys (default and easiest).
  - **SSE-KMS:** AWS Key Management Service keys (more control, audit trails).
  - **SSE-C:** Customer-provided keys.
  - **Client-Side Encryption:** Encrypt data yourself before uploading.

Common Use Cases



Summary

S3 is the foundation for storage in AWS.

It's **secure, durable, and incredibly scalable**.

Key concepts are **Buckets** (containers) and **Objects** (files).

Use **Storage Classes** and **Lifecycle Policies** to optimize costs.

Programmatic S3 Access with the AWS .NET SDK

A Developer's Guide to the Amazon.S3 **Namespace**

What is the AWS SDK for .NET?

Official .NET Library: A set of .NET assemblies that simplifies building .NET applications that use AWS services.

Abstraction Layer: Handles low-level details like signing requests, retries, and error handling.

Two Flavors:

.NET 3.5 & .NET Framework: The original "master" SDK.

.NET Standard 2.0 (Modern): The focus for new development. Supports .NET Core, .NET 5/6/7/8+, and .NET Framework 4.6.1+.

Core Component: The AWSSDK.S3 NuGet package.

Core Namespace: Amazon.S3

The primary namespace containing all S3-specific classes, interfaces, and enumerations.

IAmazonS3: The main service interface. This is what you code against for testability.

AmazonS3Client: The concrete implementation of IAmazonS3. Your entry point to the API.

Request/Response Pattern: For every operation, there is a dedicated **Request** class (e.g., PutObjectRequest) and a **Response** class (e.g., PutObjectResponse).

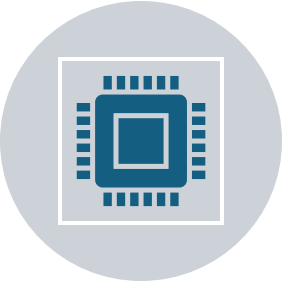
Demo – S3 File Explorer

Amazon DynamoDB with .NET

What is Amazon DynamoDB?

Fully managed NoSQL database.

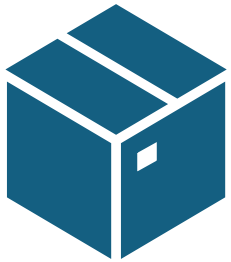
Key Concepts: Tables, Items, Attributes, Primary Key (Partition + Sort Key).



Benefits: Serverless, single-digit millisecond performance, automatic scaling.

When to use it: High-traffic web apps, serverless applications, any app needing predictable performance at scale.

.NET SDK for DynamoDB



NuGet
Packages: AWSSDK.DynamoDBv2

Two Access Styles:

Low-Level
Client: AmazonDynamoDBClient (uses PutItemRequest, GetItemRequest).

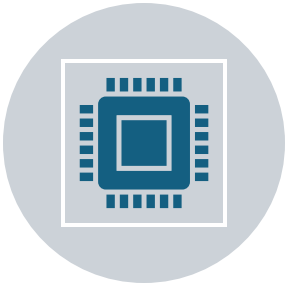
Document Model: Table and Document classes (flexible, but less type-safe).

Object Persistence Model
(Recommended): DynamoDBContext (maps C# classes to DynamoDB tables).

Demo – DynamoDB CRUD

DynamoDB Accelerator - DAX

What is DAX?



Fully managed, in-memory cache for DynamoDB.

Benefit: Provides microsecond response times for eventually consistent reads.

How it works: Sit it in front of your DynamoDB table. App talks to DAX, DAX talks to DynamoDB.

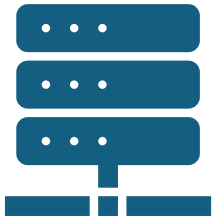
Use Case: Read-heavy workloads that need extreme speed (e.g., popular product pages, gaming leaderboards).

RDS for SQL Server

What is Amazon RDS?

Relational Database Service - Managed SQL Database.

Why use it over EC2? Handles provisioning, patching, backup, recovery, scaling, failover.

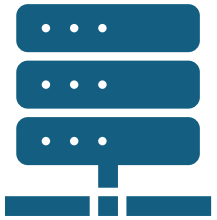


Supports SQL Server, MySQL, PostgreSQL, MariaDB, Oracle.

Key Considerations for RDS

Security Groups: Your app's EC2 instance/security group must be allowed to connect to the RDS instance's security group on port 1433.

IAM Authentication: (Optional) You can use IAM roles to generate database authentication tokens instead of passwords.



High Availability: Explain Multi-AZ deployment.

Lab – Build a .NET app with S3

Summary – Day 1

AWS Fundamentals & .NET Integration

Session 1: Introduction to AWS & .NET SDK (2 hrs)

- AWS Global Infrastructure (Regions, AZs)
- AWS SDK for .NET (C#) setup & configuration
- IAM (Identity & Access Management) for .NET apps
    - Roles, Policies, and .NET SDK integration

Hands-on Lab:

Deploy a .NET Core app using AWS SDK for IAM access

Session 2: Compute Services for .NET (2 hrs)

- AWS Lambda with .NET
    - Writing C# Lambda functions
    - Triggers (API Gateway, S3, DynamoDB)
- EC2 for .NET Applications
    - Deploying ASP.NET Core on EC2

Hands-on Lab:

- Build & deploy a serverless .NET API with Lambda + API Gateway

Session 3: Storage & Database Services (3 hrs)

- Amazon S3 – Storing & retrieving files from .NET
- DynamoDB – NoSQL with .NET SDK
    - CRUD operations in C#
    - DynamoDB Accelerator (DAX)
- RDS for SQL Server – Running managed SQL with .NET

Hands-on Lab:

- Build a .NET app with DynamoDB & S3 integration

Feedback