AWS Serverless

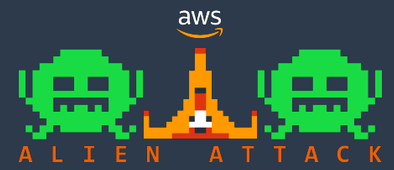
Prática do Dev ao SysAdmin

Fabian Da Silva
Senior Solutions Architect, AWS Partner Network
@fabianmartins

Oct, 14, 2020

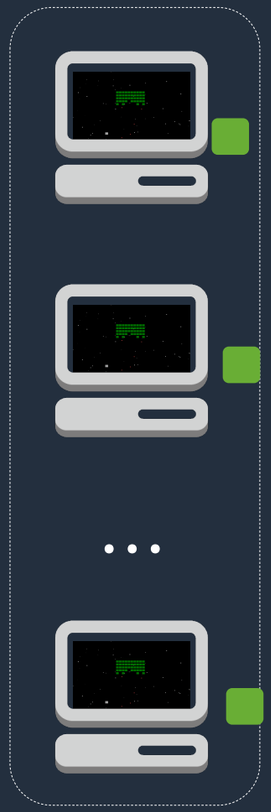
Falando de Serverless

Porque começamos com Alien Attack?



aws

Padrão Produtor-Consumidor



Gamers
(Produtores)

Gerente
(Consumidor)

Casos de uso relacionados

- Acompanhamento de vendas em tempo-real
- Detecção tem tempo-real de eventos a partir de beacons em lojas de deparmento (Marketing/promoções em near real-time)
- Tratamento de transações financeiras em tempo-real (tendências, prevenção à fraudes...)
- Processamento de eventos de dispositivos em tempo-real (IoT, Mobile)
- Monitoramento de pacientes em hospitais em tempo-real
- Rastreamento de ativos/cargas em tempo-real
- Cenário de gamificação (onde um scoreboard viria a calhar?)
- ...

Requisitos da aplicação

Segurança

(RBAC)

Processamento em tempo-real

Armazenamento dos dados para uso futuro

Minimizar o TCO com Infraestrutura

Cloud Economics

Custo

Quanto paguei neste mês com infra-estrutura?

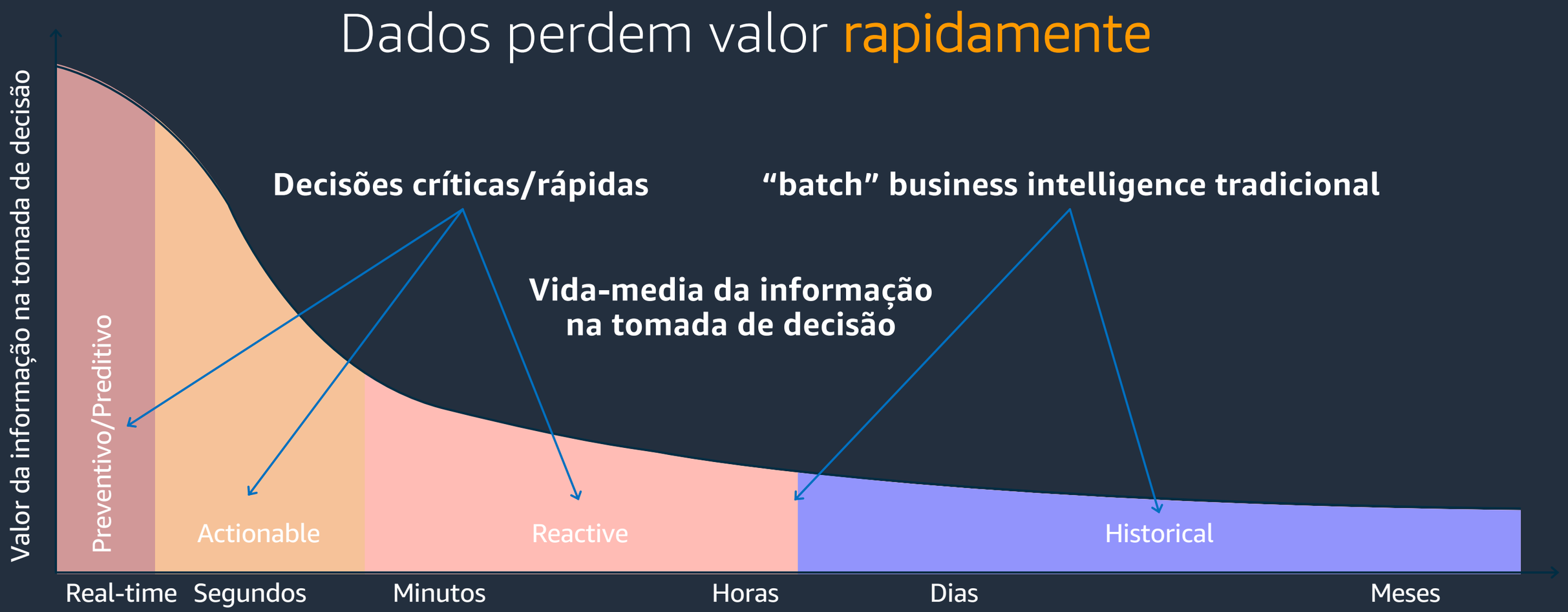
TCO (Total Cost of Ownership)

Qual o custo total para manter a minha infra-estrutura executando?

- Energia elétrica
- Rede
- Armazenamento
- Processamento
- Licenças
- Horas de suporte/manutenção
- Horas do time de Dev/Test/QA
- Custo de governança e compliance
- Quanto pago de idle?
- … em todos os ambientes

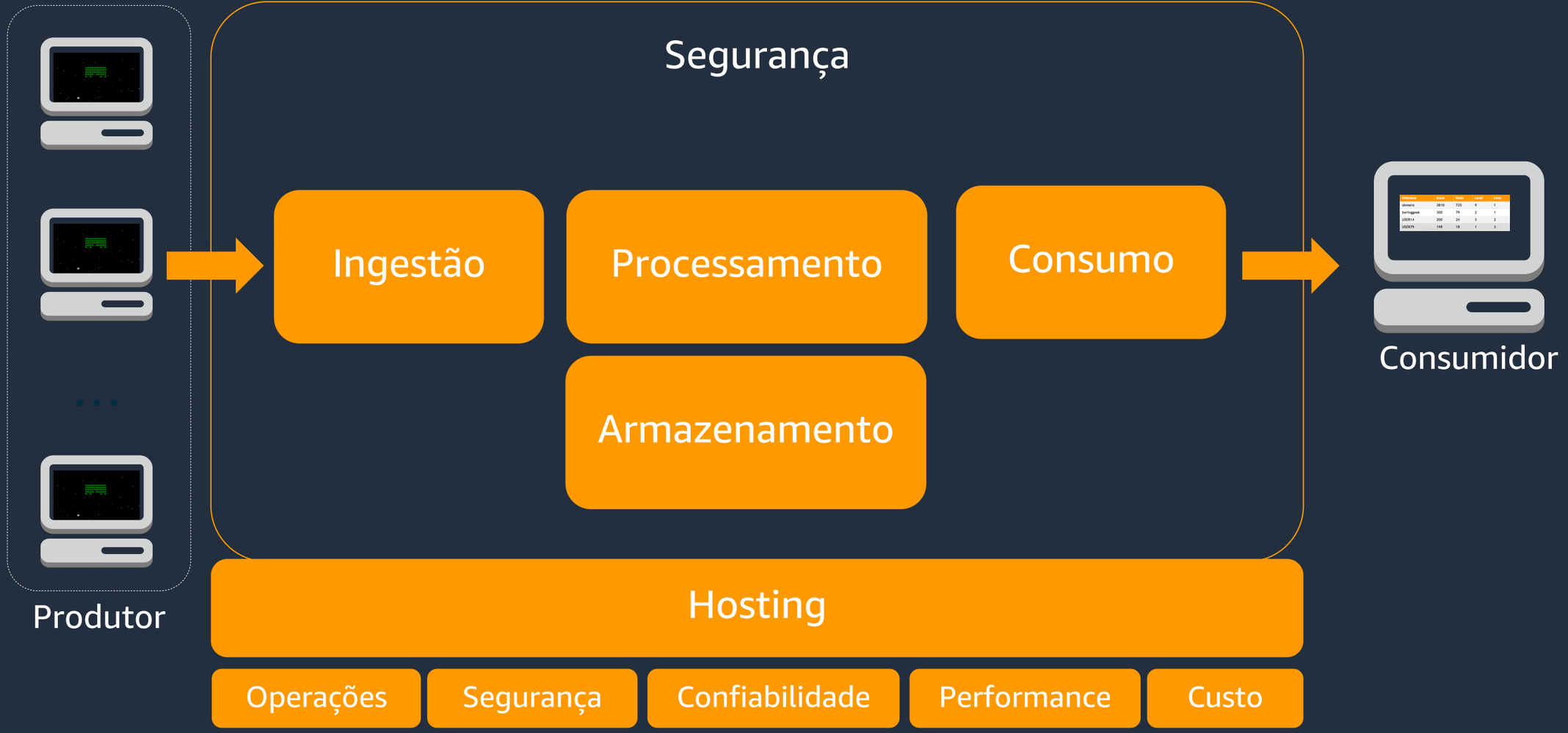
Para comparar Cloud com on-premises, você precisa saber o TCO on-premises, e em geral não se sabe (excessiva alocação de custos em "*overhead*")

O que significa “real-time”?



Source: Perishable insights, Mike Gualtieri, Forrester

Arquitetura de alto-nível padrão



Vários caminhos...

Mas, afinal, o que é Serverless?

Sem infra-estrutura para provisionar,
Sem gerenciamento

Escalabilidade automática

Computação sem Servidores

Comunicação
Processamento
Armazenamento

Sem que eu precise gerenciar
(instalar, atualizar, escalar...)

Pague pelo valor adicionado

Altamente disponível e seguro

(Infraestrutura Virtualizada Tradicional + API + Padrões de Automação) X Engenheiros brilhantes + Camada de Abstração = Serverless

aws

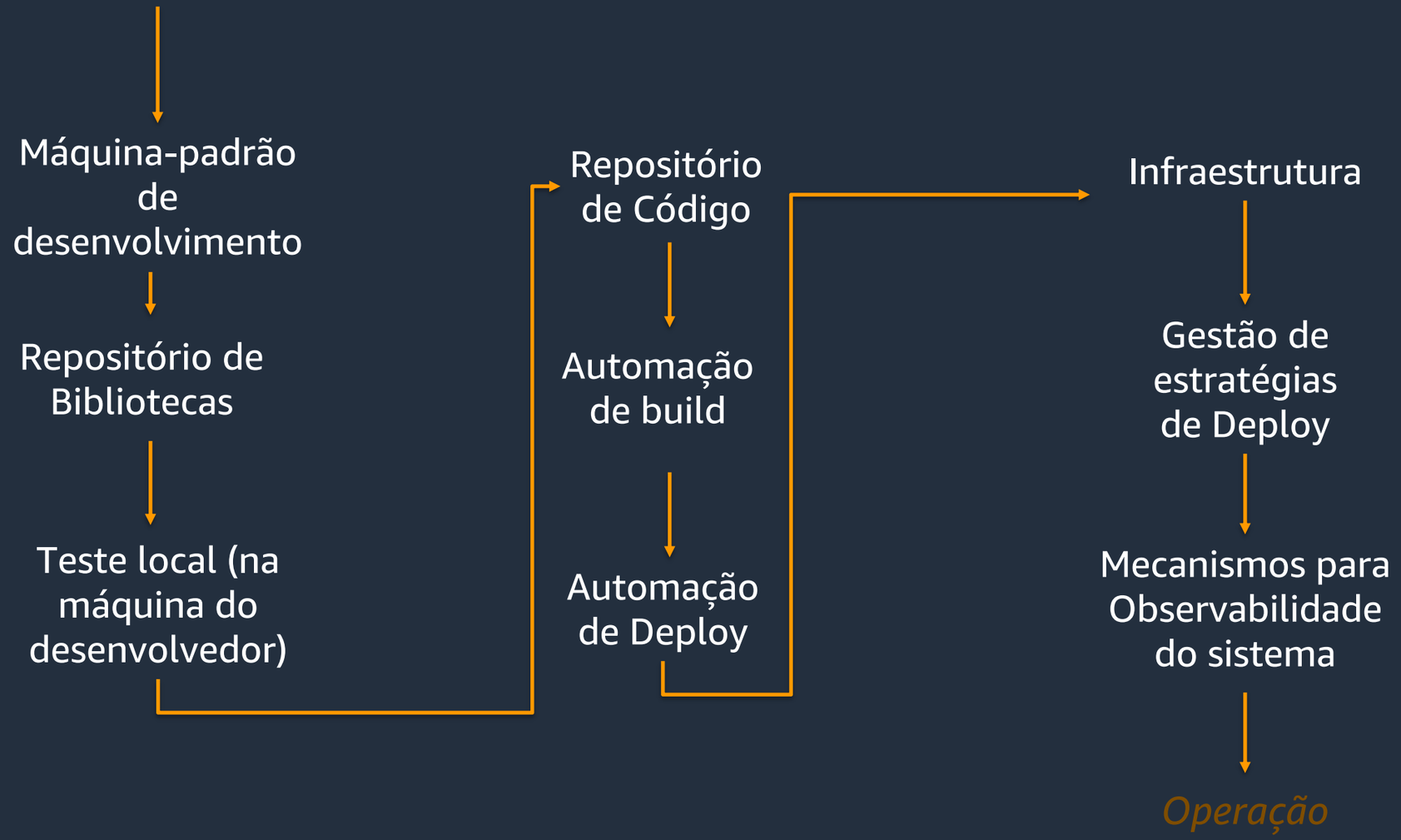
AWS Alien Attack

Como você construiria esse sistema?

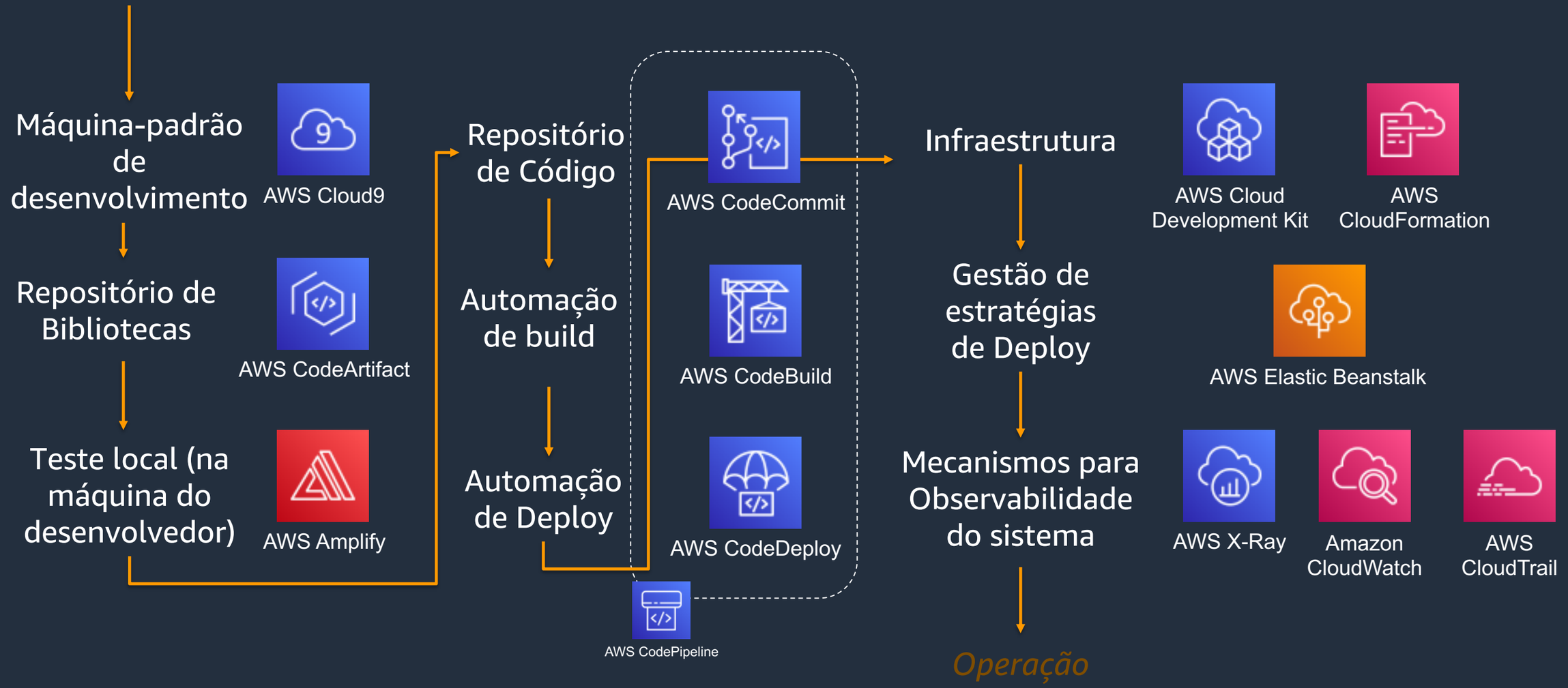
Desenvolvimento
Infra-estrutura
Operação

Desenvolvimento do Sistema

Todo time de desenvolvimento precisa...



Todo time de desenvolvimento precisa...



AWS Cloud9
Cloud-based integrated development environment

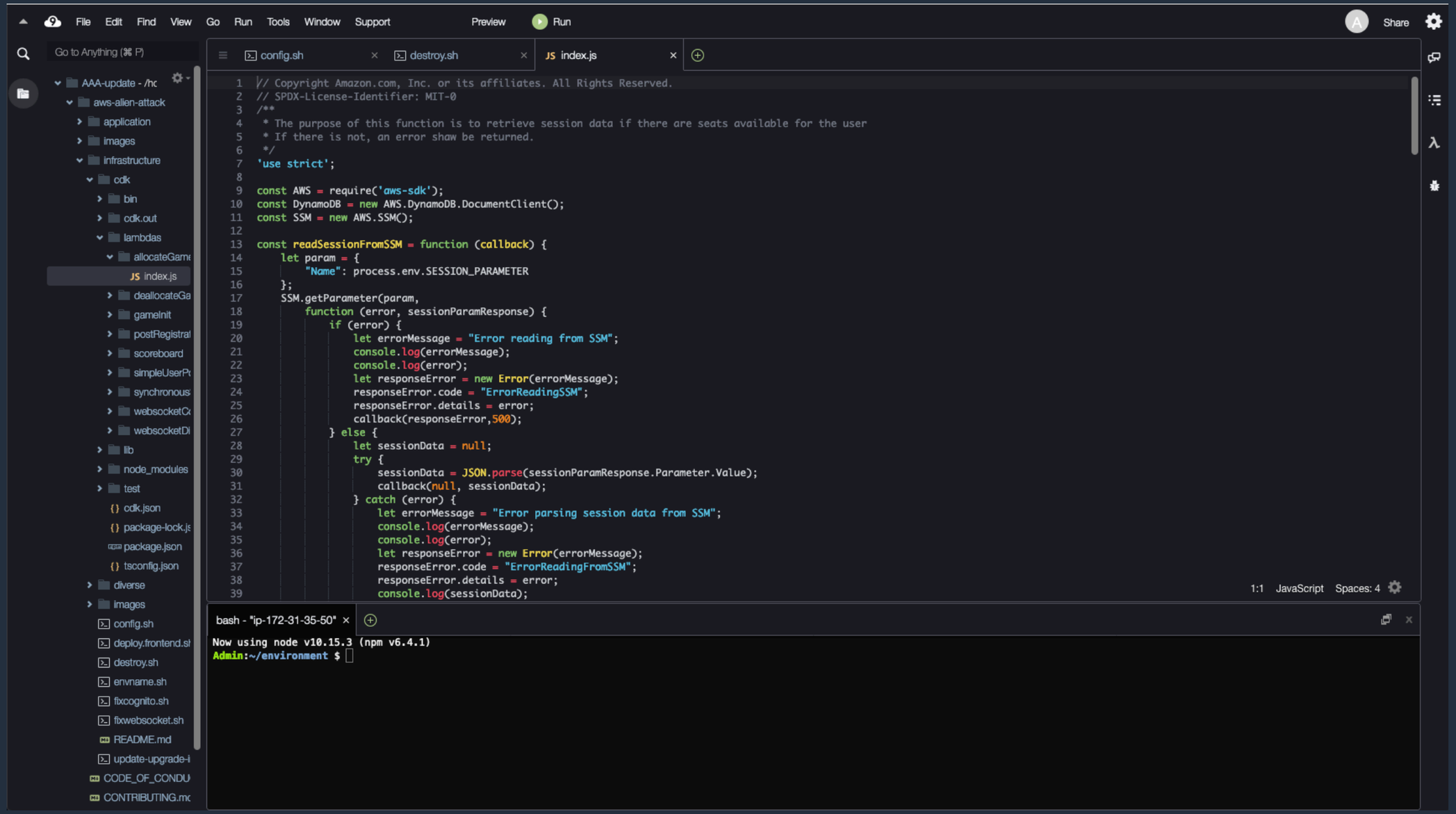
Programe usando o browser

IDE completa

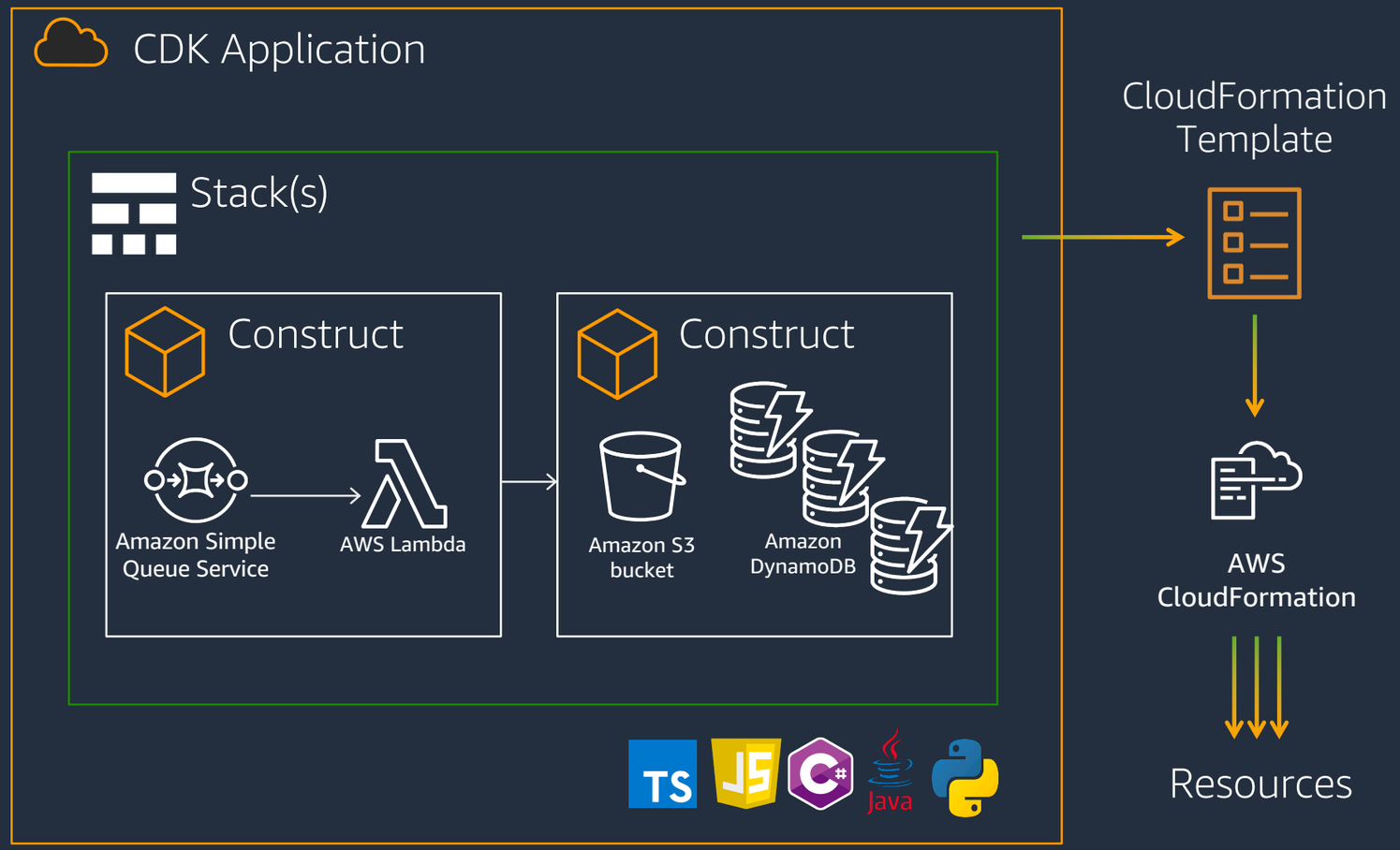
Construa aplicações serverless

Colabore em tempo-real

AWS Cloud9



AWS Cloud Development Kit (CDK)



CDK Sample Code

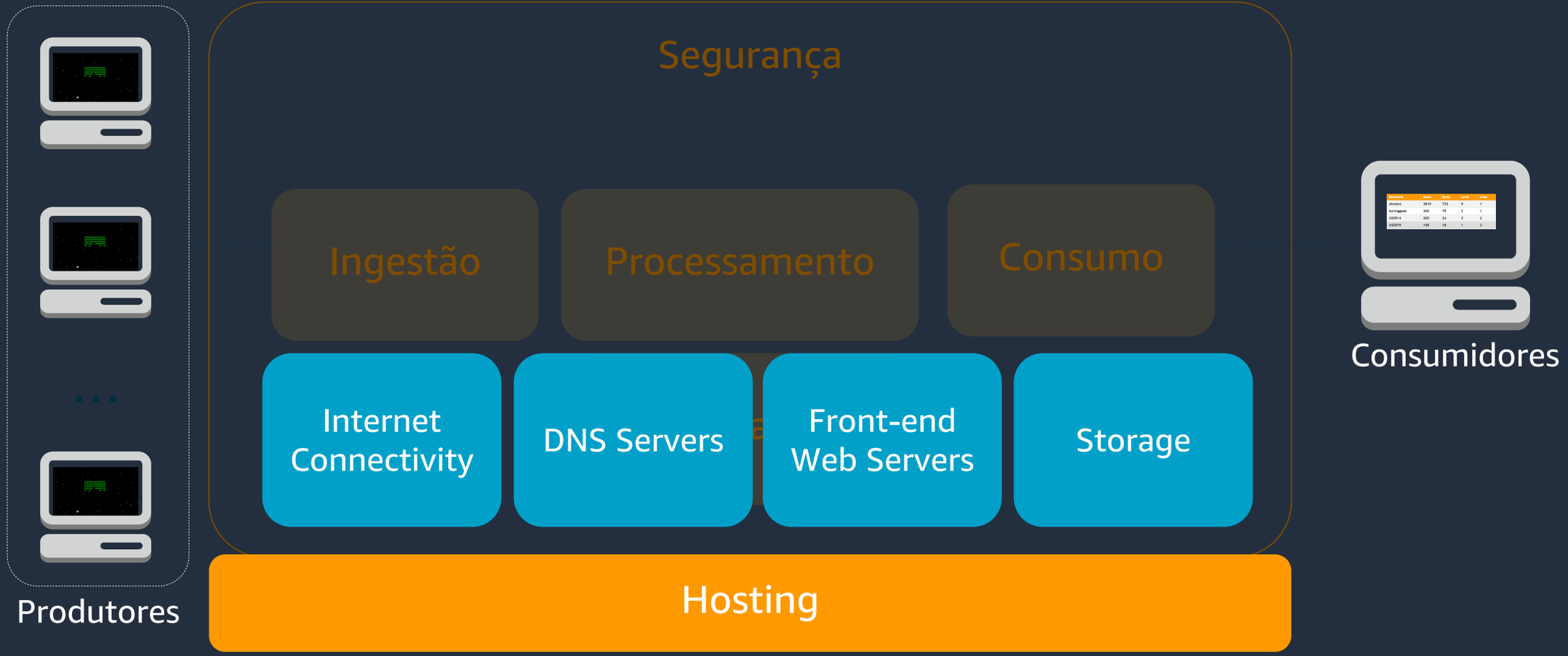
```
export class ECSStack extends cdk.Stack {

public readonly theAPI: apigV1.RestApi
public readonly cluster: ecs.Cluster
public readonly loadBalancer: elbv2.NetworkLoadBalancer
public readonly vpcLink: apigV1.VpcLink

constructor(scope: cdk.App,
            id: string, props?:  cdk.StackProps)
{
  super(scope, id, props);

  const vpc =
     new ec2.Vpc(this, 'ecs_Vpc', { maxAzs: 2 });

  this.cluster = new ecs.Cluster(this, 'ecsCluster', {
    clusterName: "ecs-Cluster",
    ContainerInsights: true,
    vpc
  });
```

```
  this.loadBalancer
     = new elbv2.NetworkLoadBalancer(this,'ecsLB’,
        { vpc, internetFacing: false });

  this.vpcLink =
   new apigV1.VpcLink( this,   'ecsVpcLink’,
       { targets: [this.loadBalancer] });

  this.theAPI =
    new apigV1.RestApi(this, "RestApi",
      { restApiName: `API ${this.stackName}`,
        endpointTypes:
           [apigV1.EndpointType.REGIONAL],
      }
    );

  const configResource =
     this.addResource( `config`, 8080 );
  const updateScoreResource =
     this.addResource( `updatescore`, 8081 );

    . . .
}
```

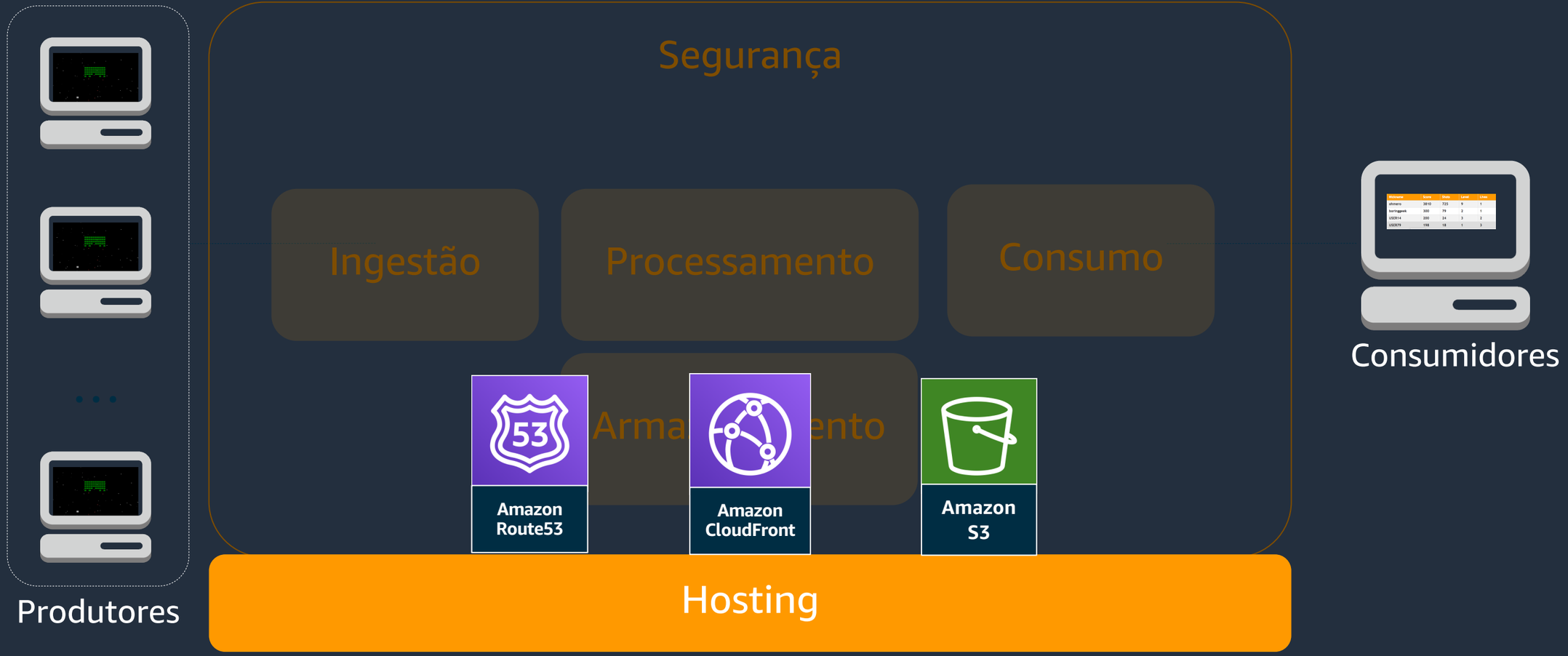

aws

Definindo a Infra-Estrutura

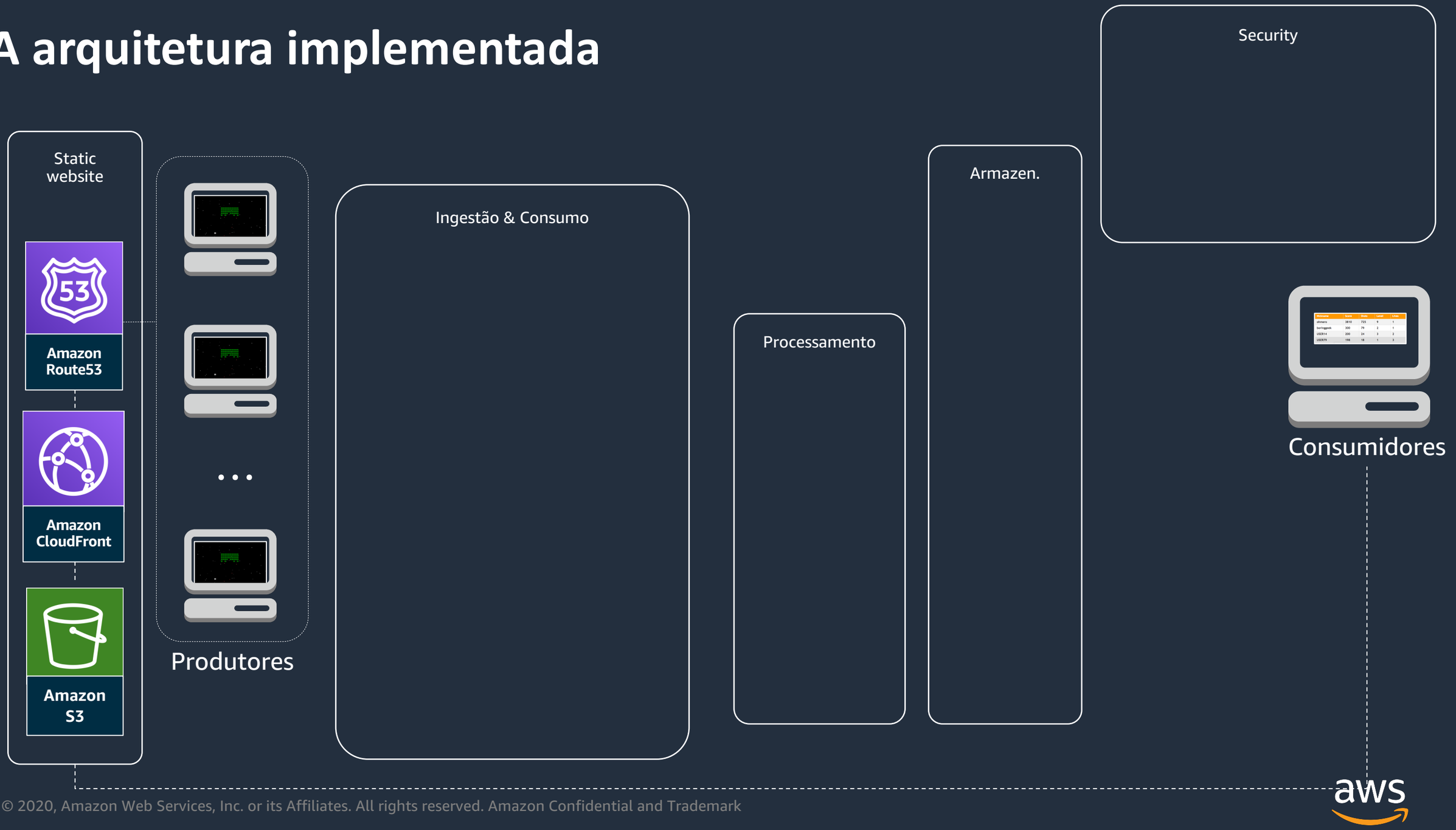
Website hosting



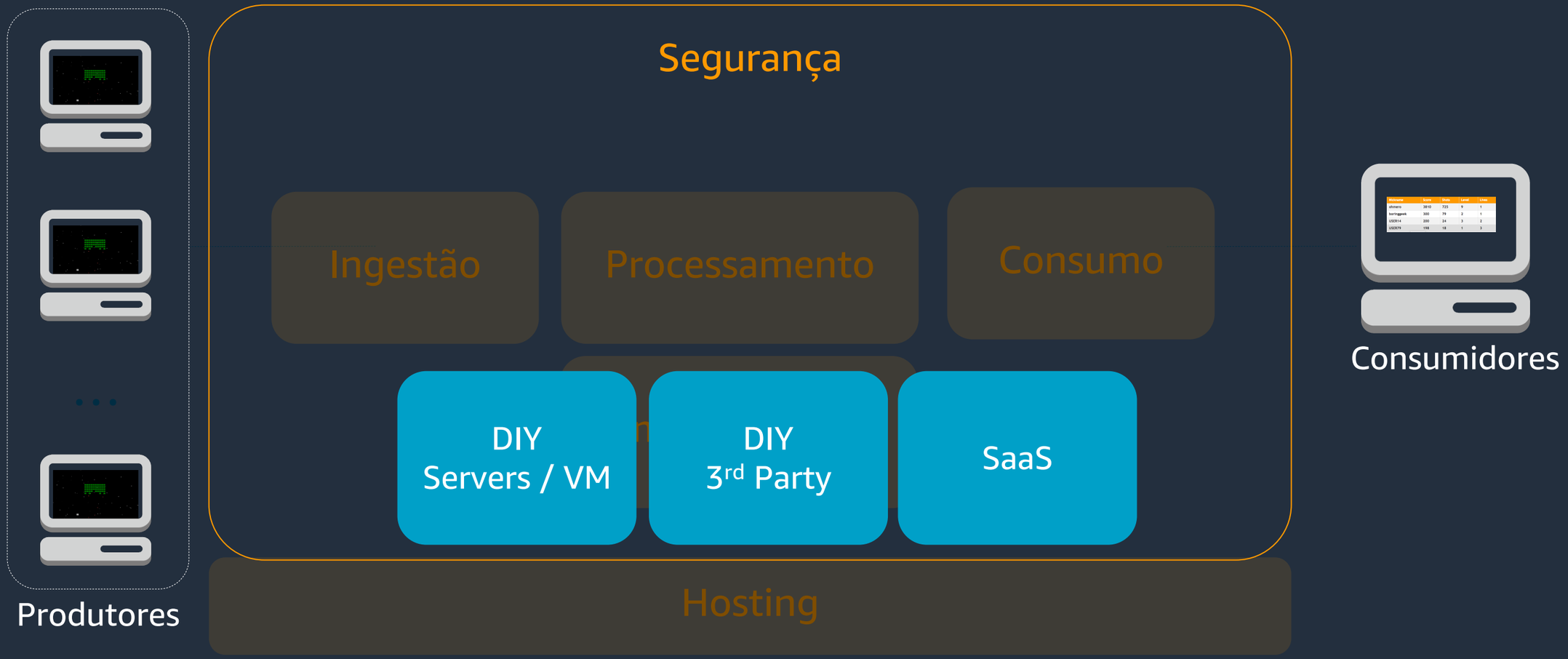
Website hosting



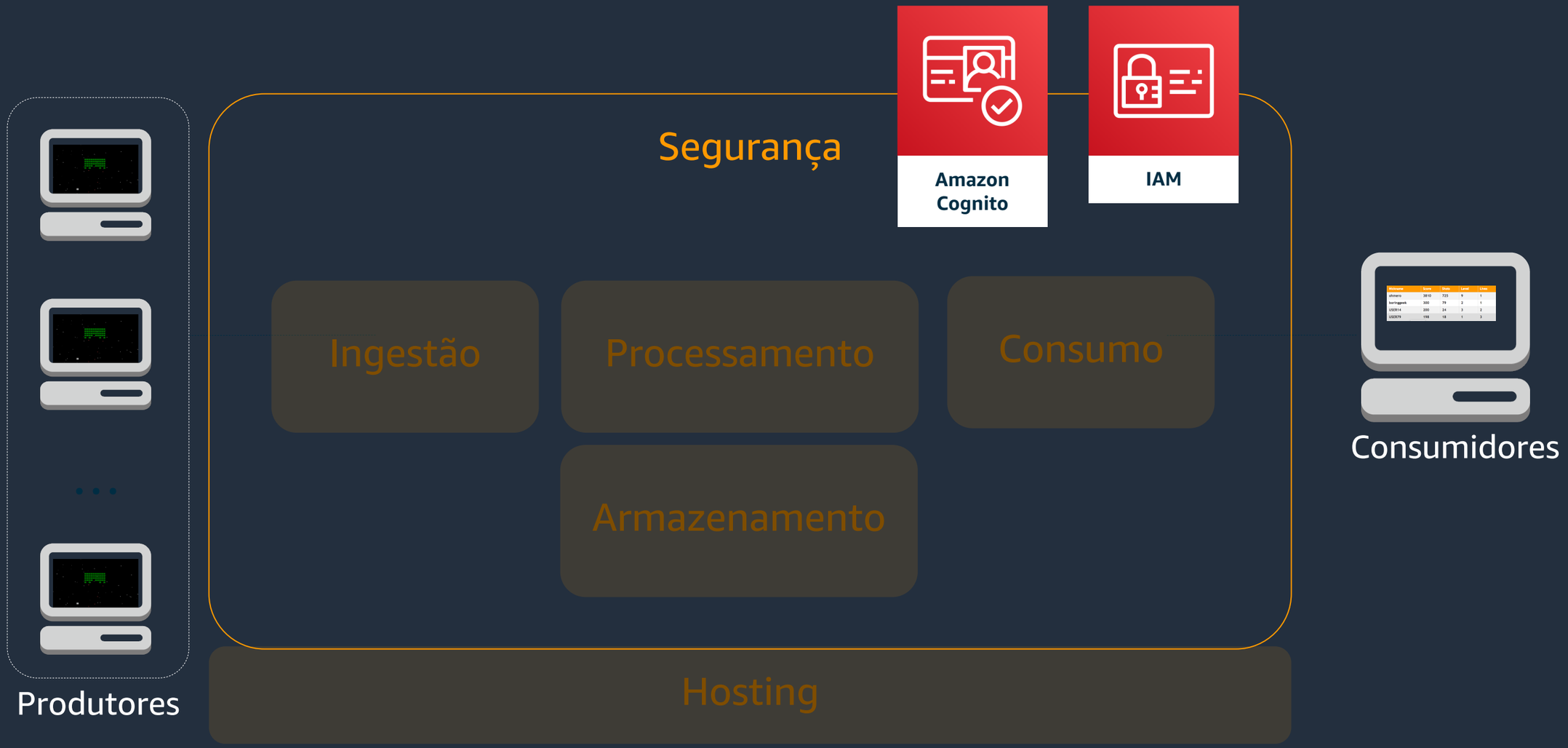
A arquitetura implementada



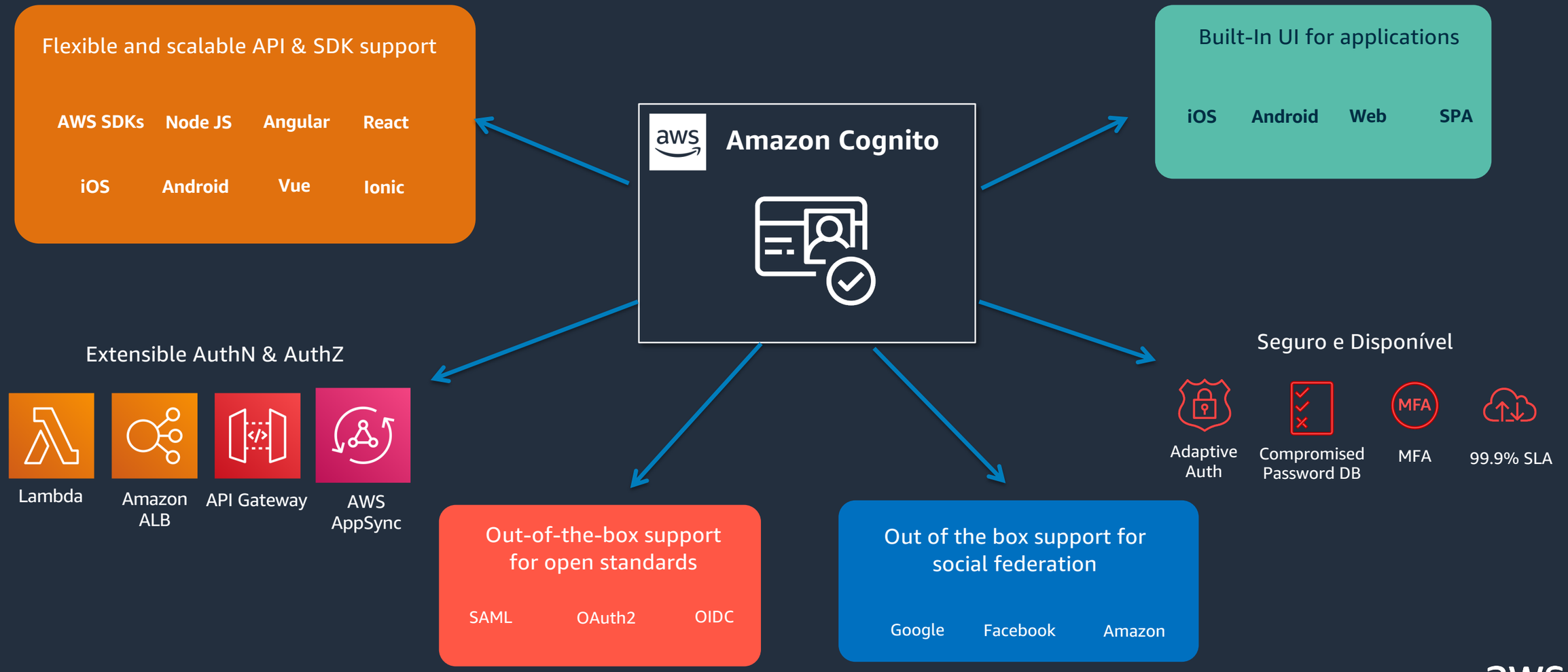
Segurança (RBAC – Role Based Access Control)



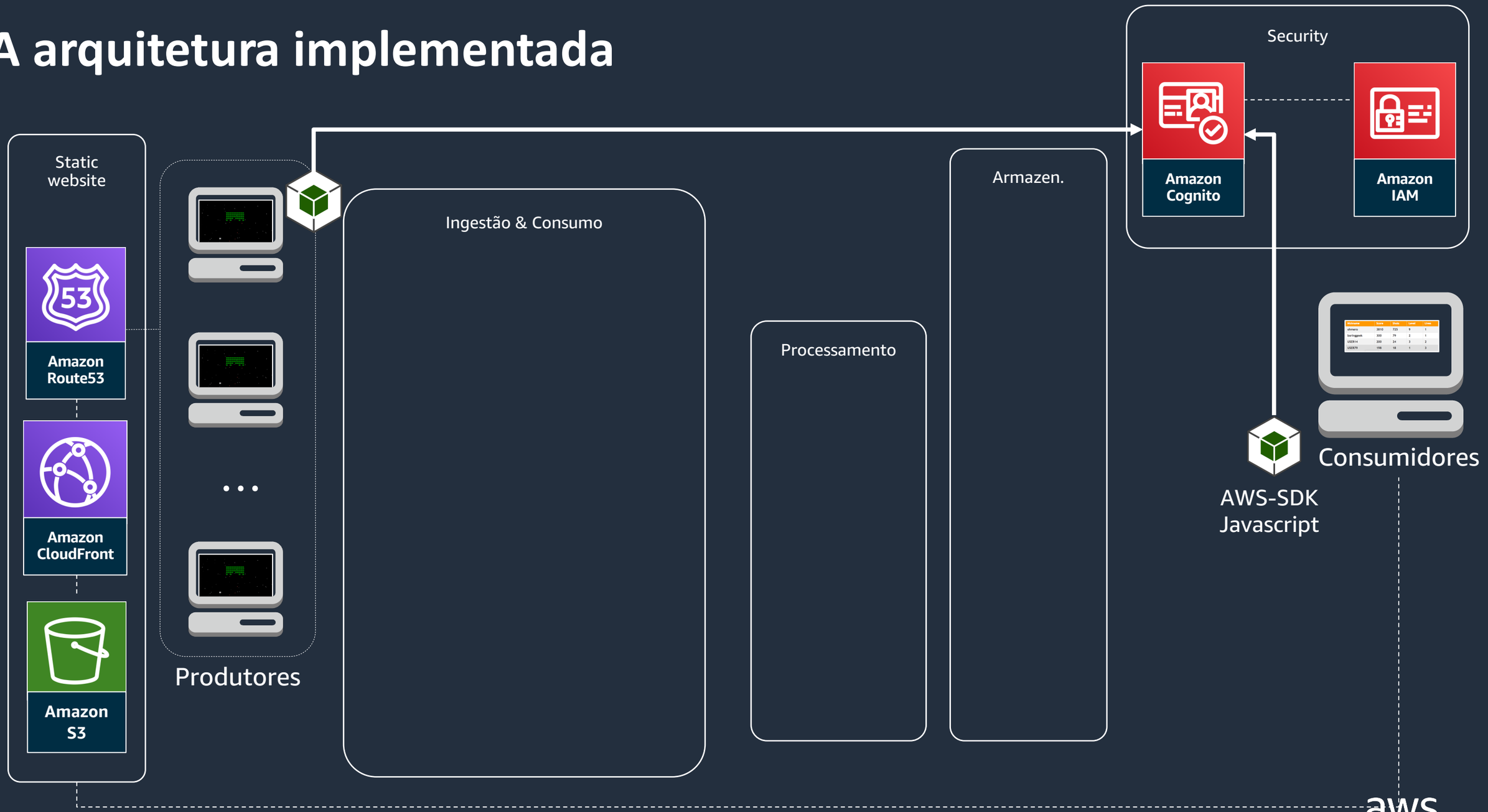
Segurança (RBAC – Role Based Access Control)



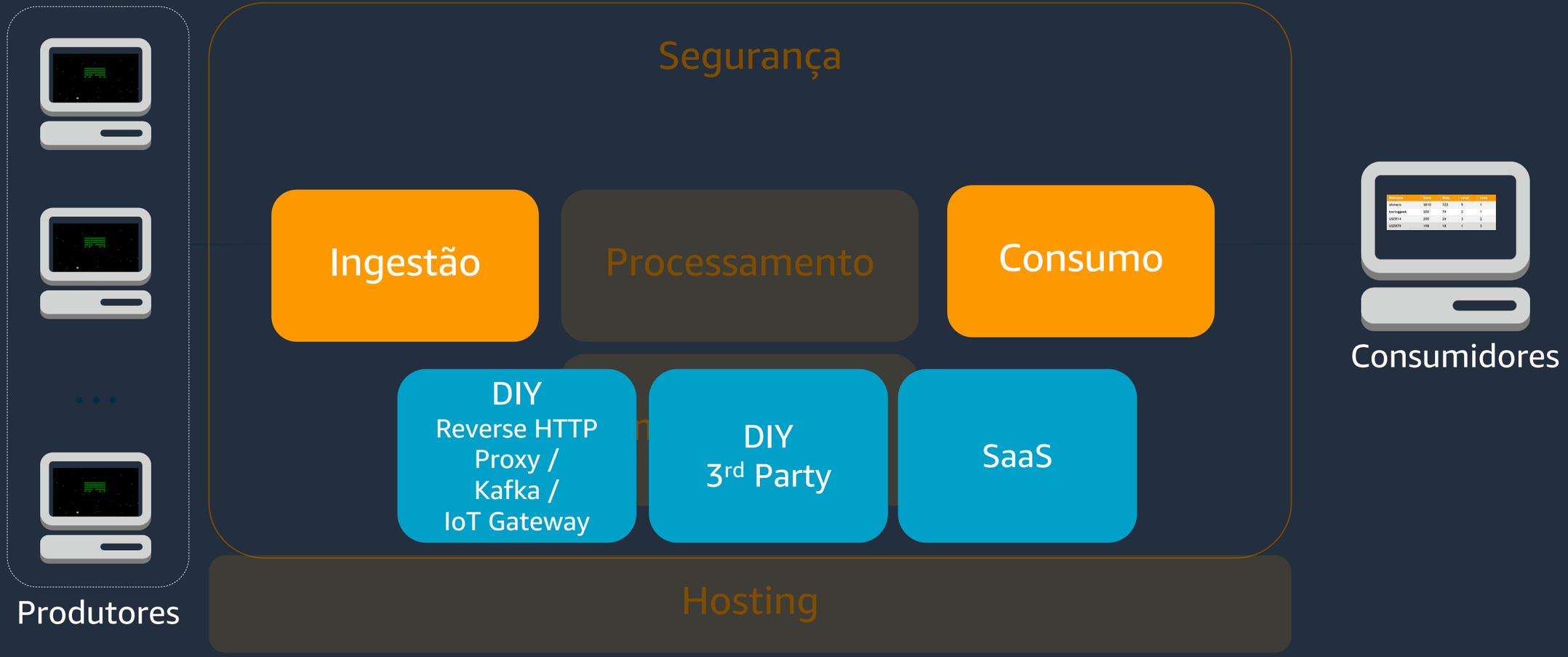
Amazon Cognito



A arquitetura implementada

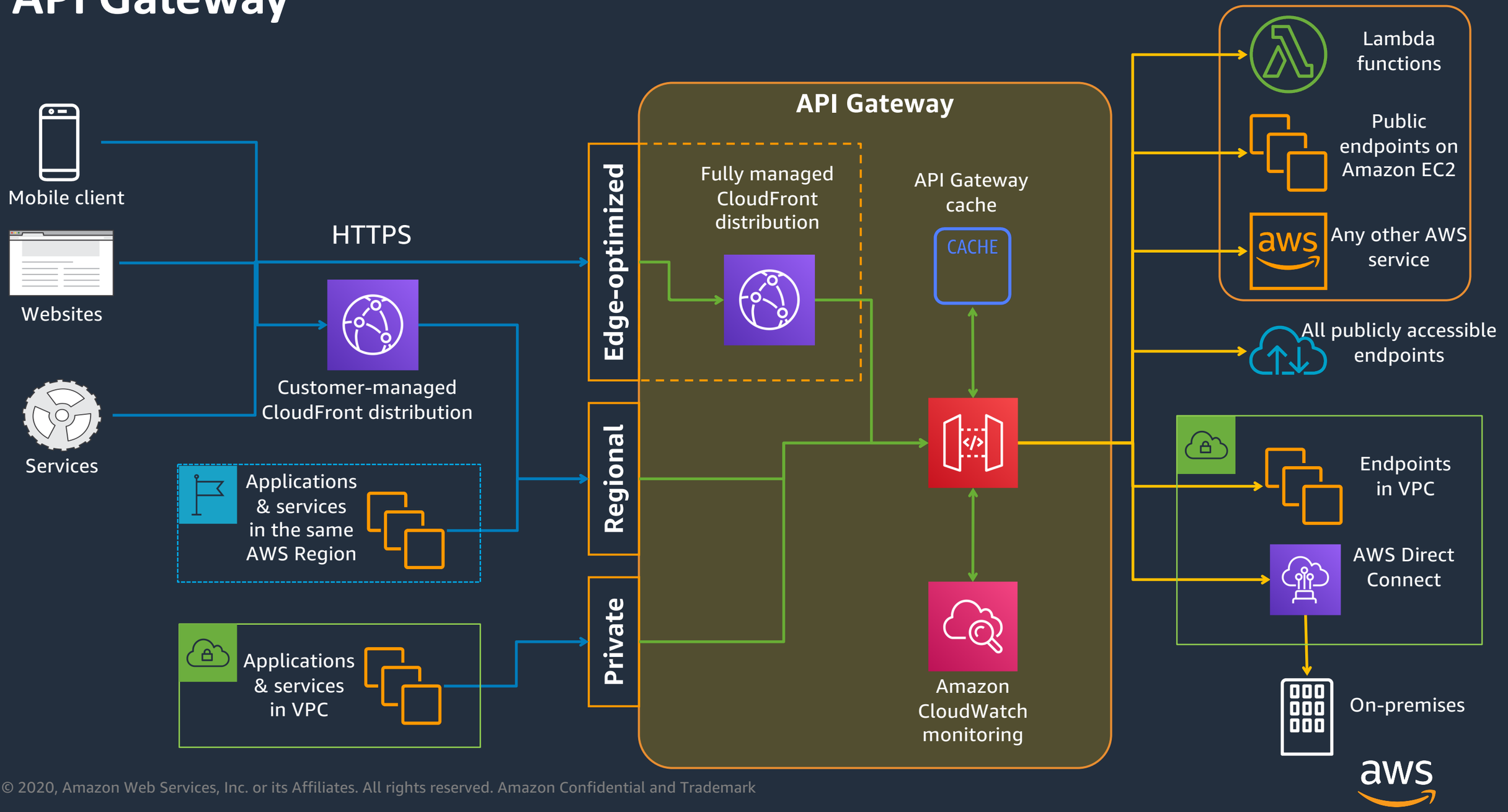


Ingestão e Consumo de Dados

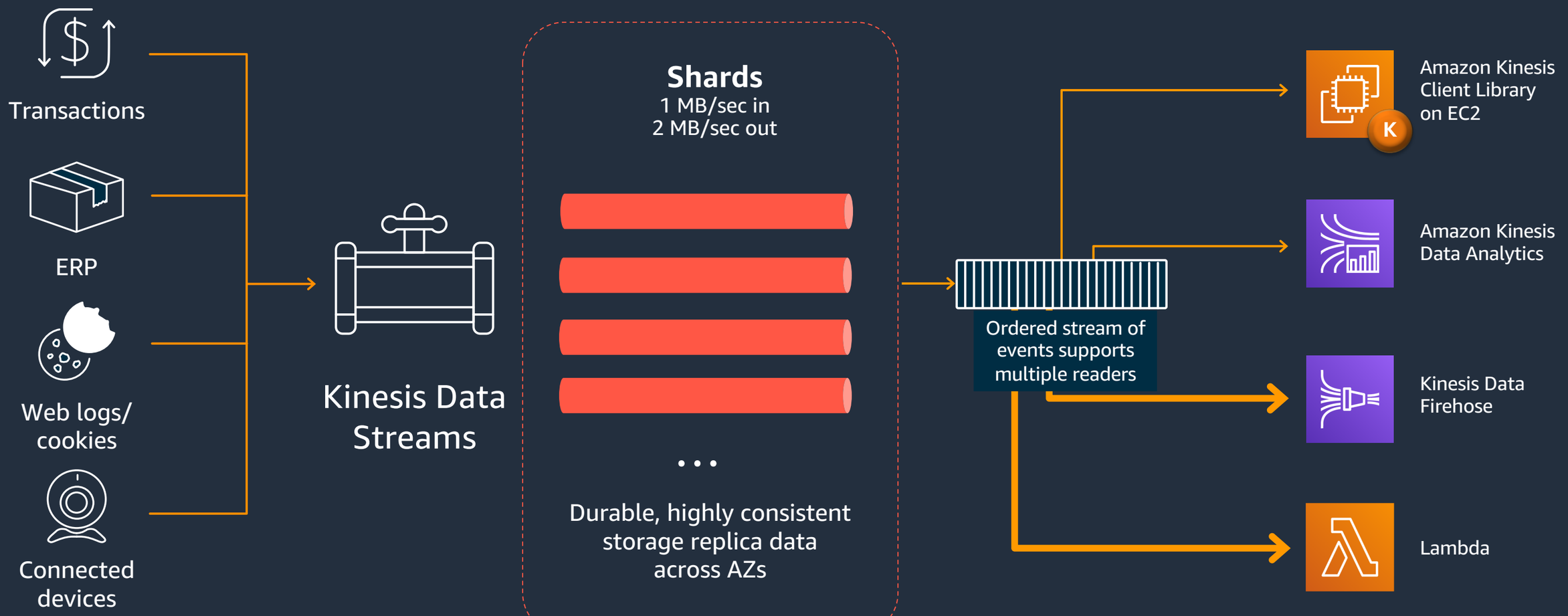


Ingestão e Consumo de Dados

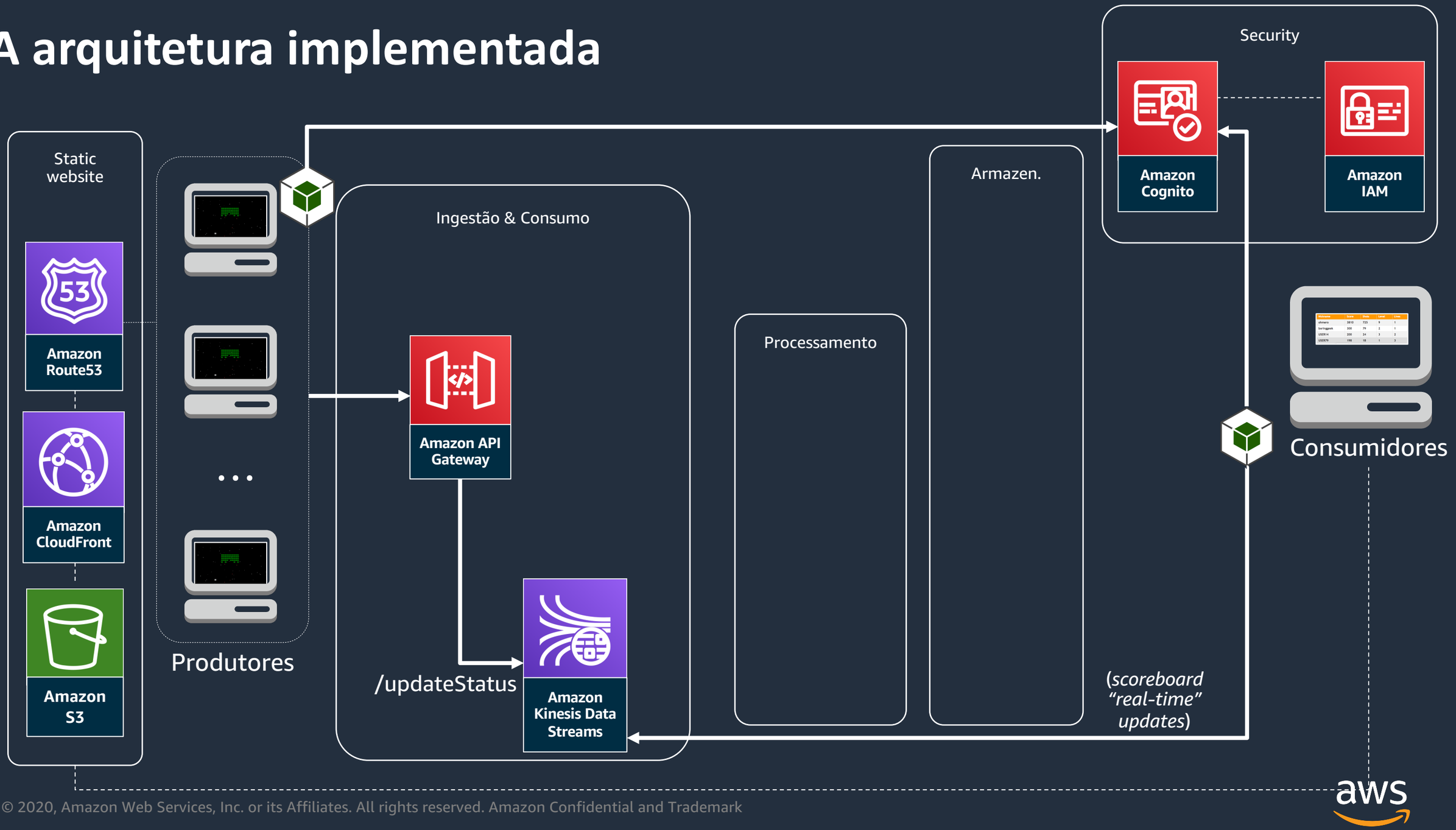
API Gateway



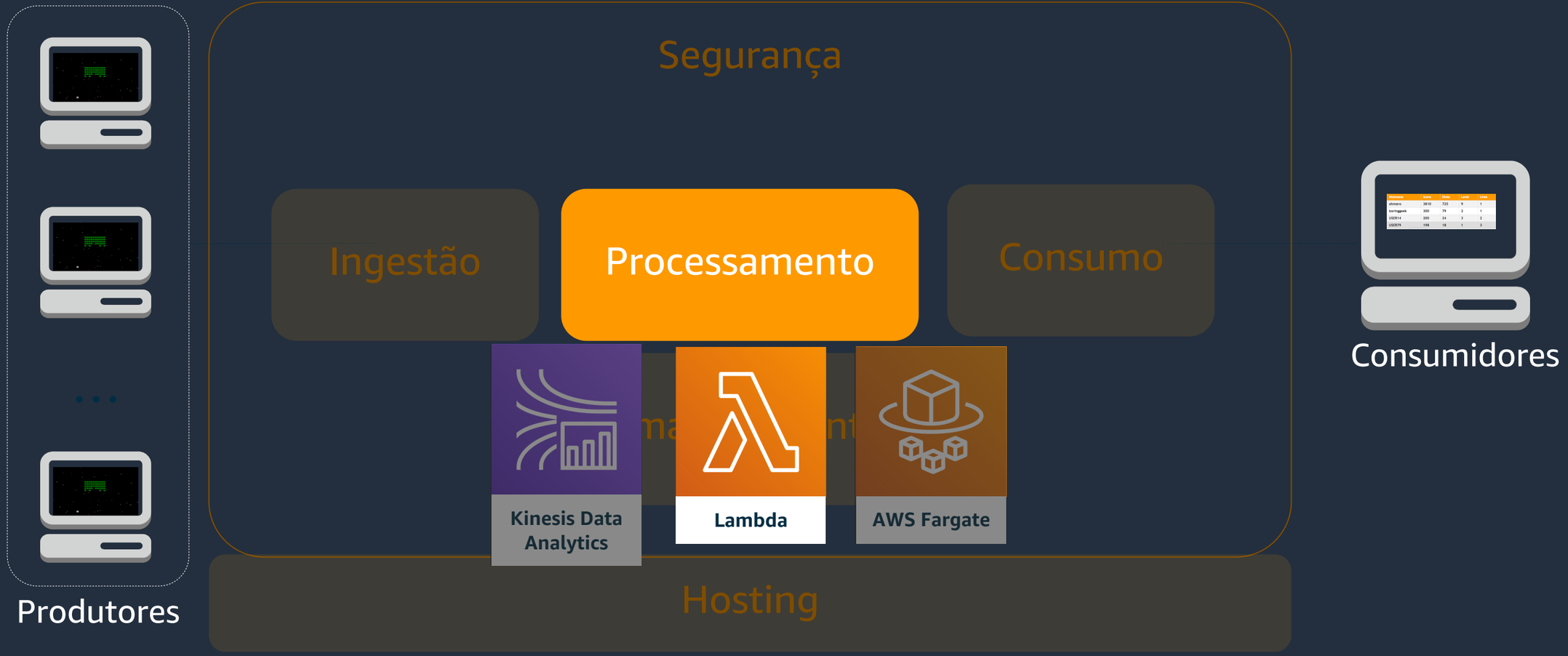
Kinesis Data Streams



A arquitetura implementada



Processamento



AWS Lambda

Event source

Changes in data state

Requests to endpoints

Changes in resource state

Function

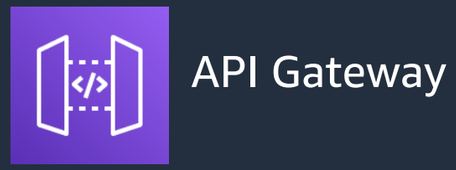
Node.js
Python
Java
.NET Core
Go
Ruby
BYOL

Services (anything)

aws

AWS Lambda Execution Model

Synchronous (push)



API Gateway

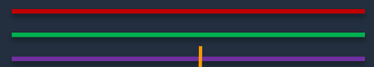
/order

Lambda function

Asynchronous (event)

Amazon SNS

Amazon S3



reqs

Lambda function

Poll-based

DynamoDB

Amazon Kinesis

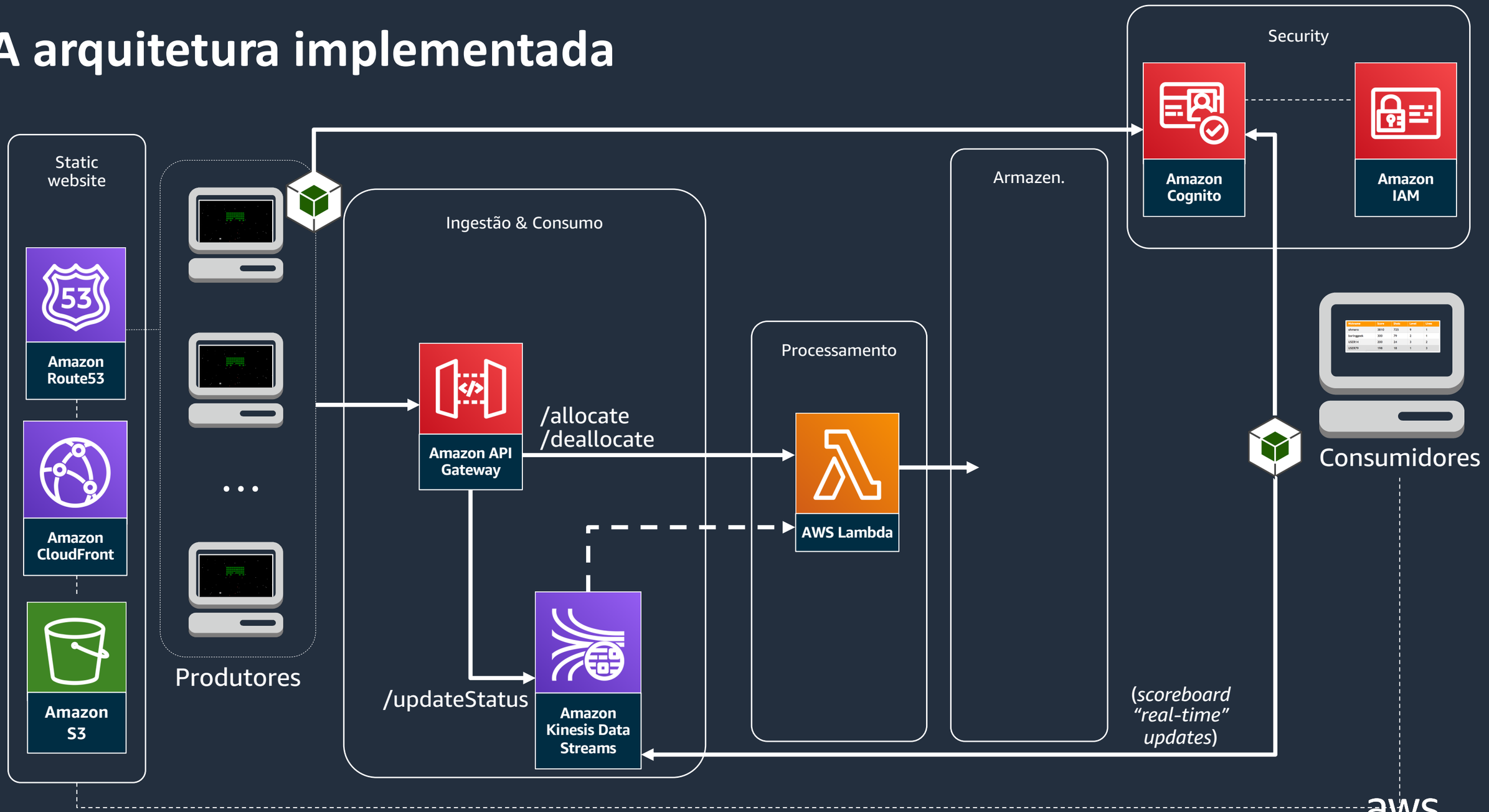
changes

Lambda service

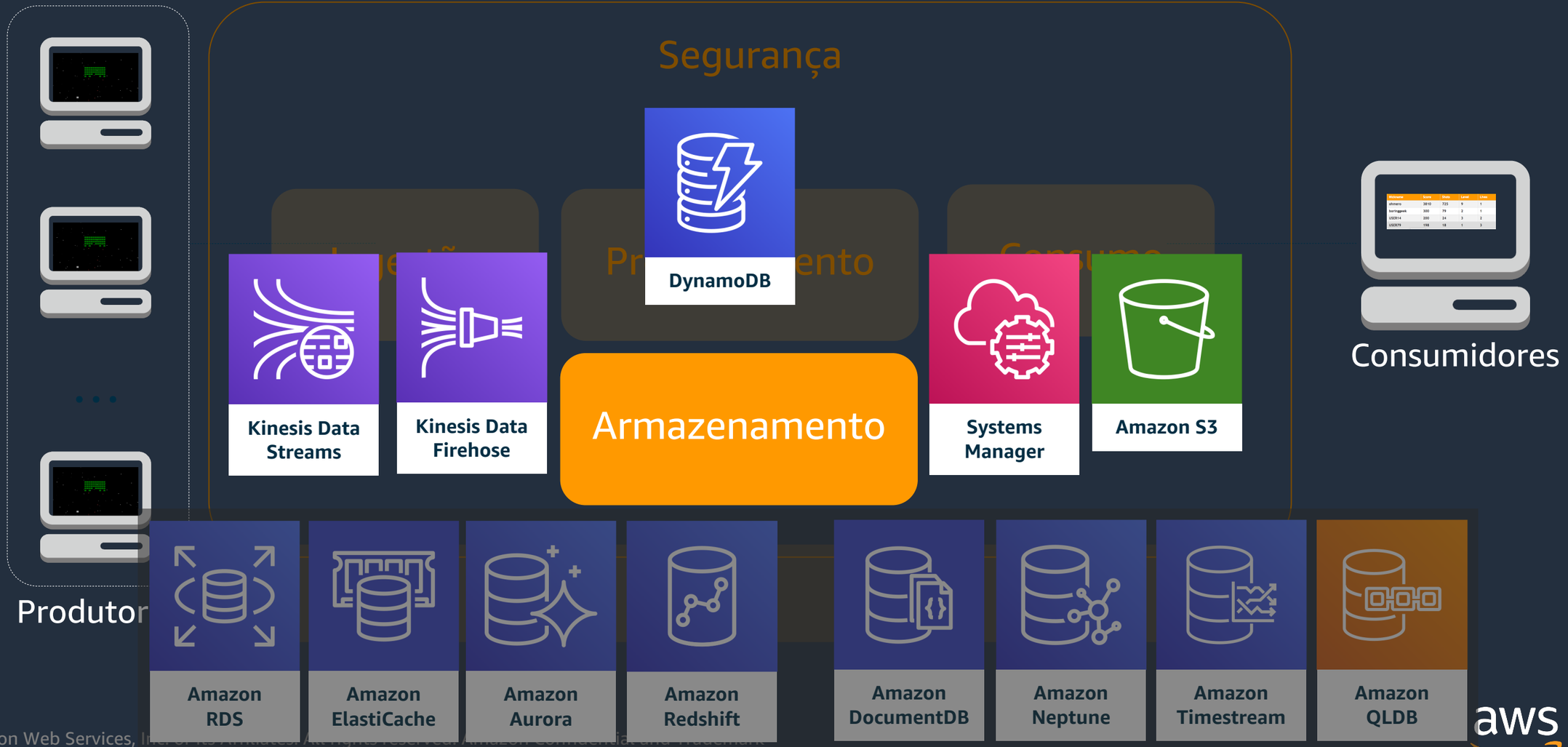
Lambda function

aws

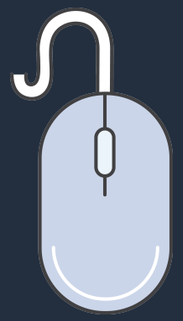
A arquitetura implementada



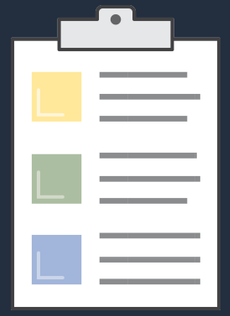
Armazenamento



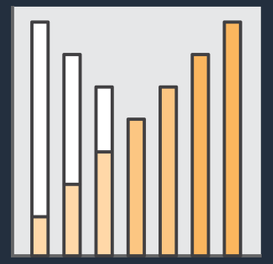
Amazon DynamoDB



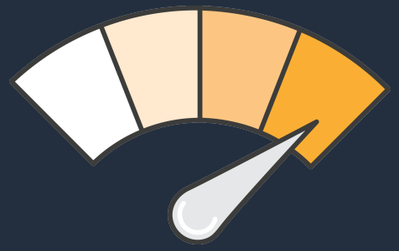
Fully managed NoSQL



Document or key-value

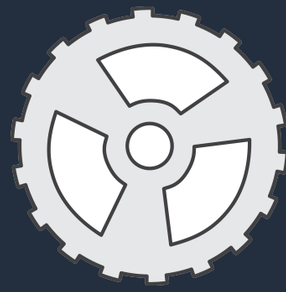


Scales to any workload



Fast and consistent

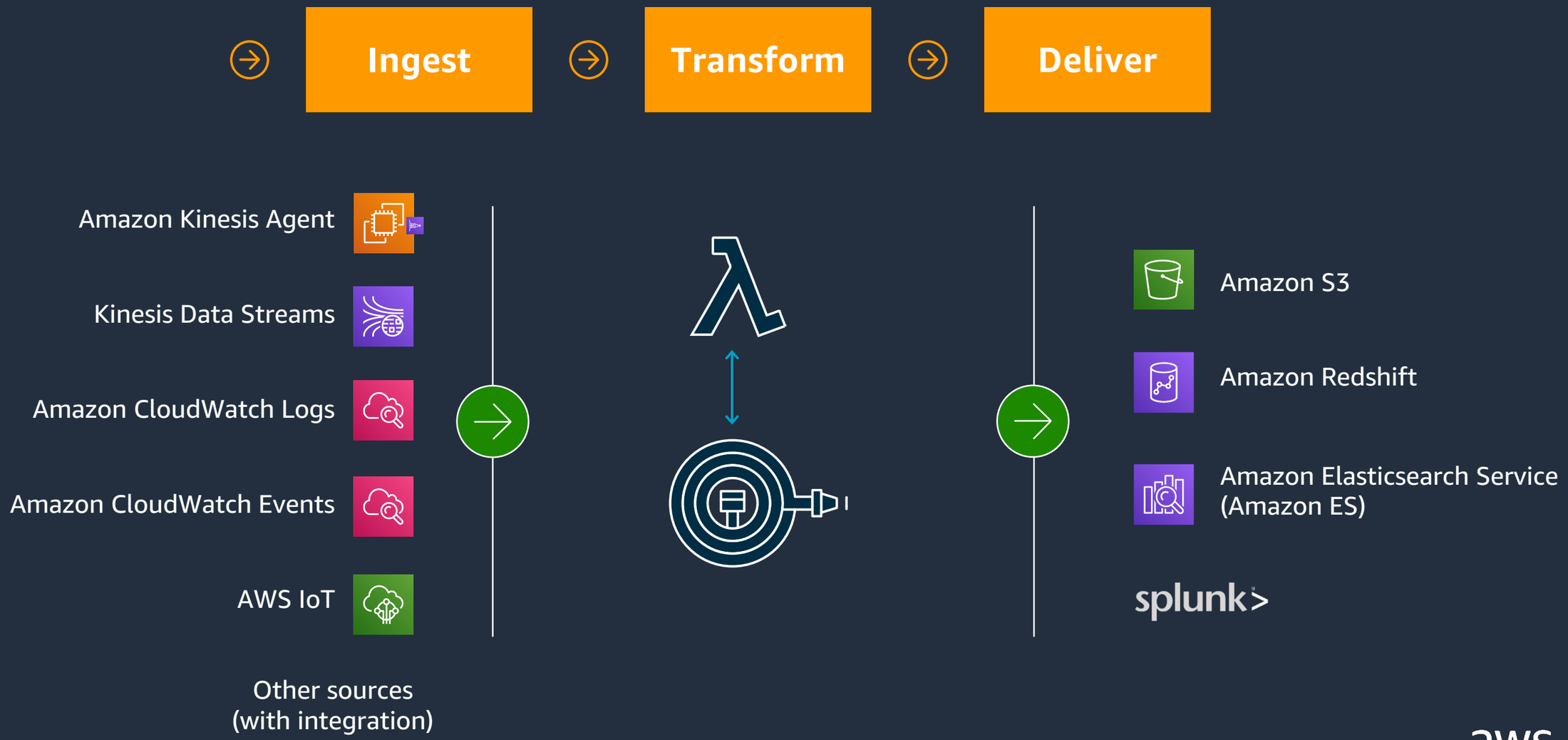
Access control



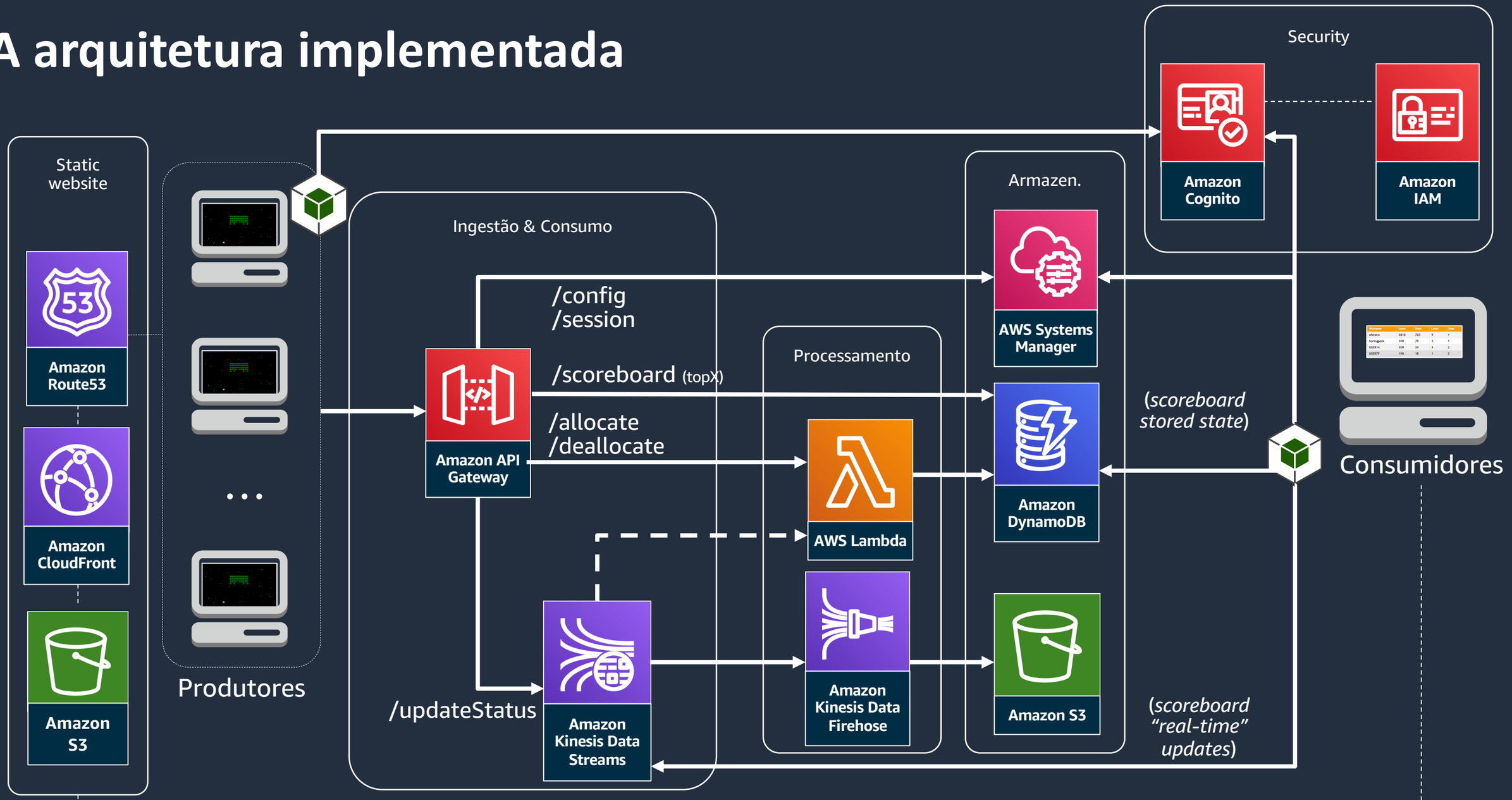
Event-driven programming

aws

Amazon Kinesis Data Firehose

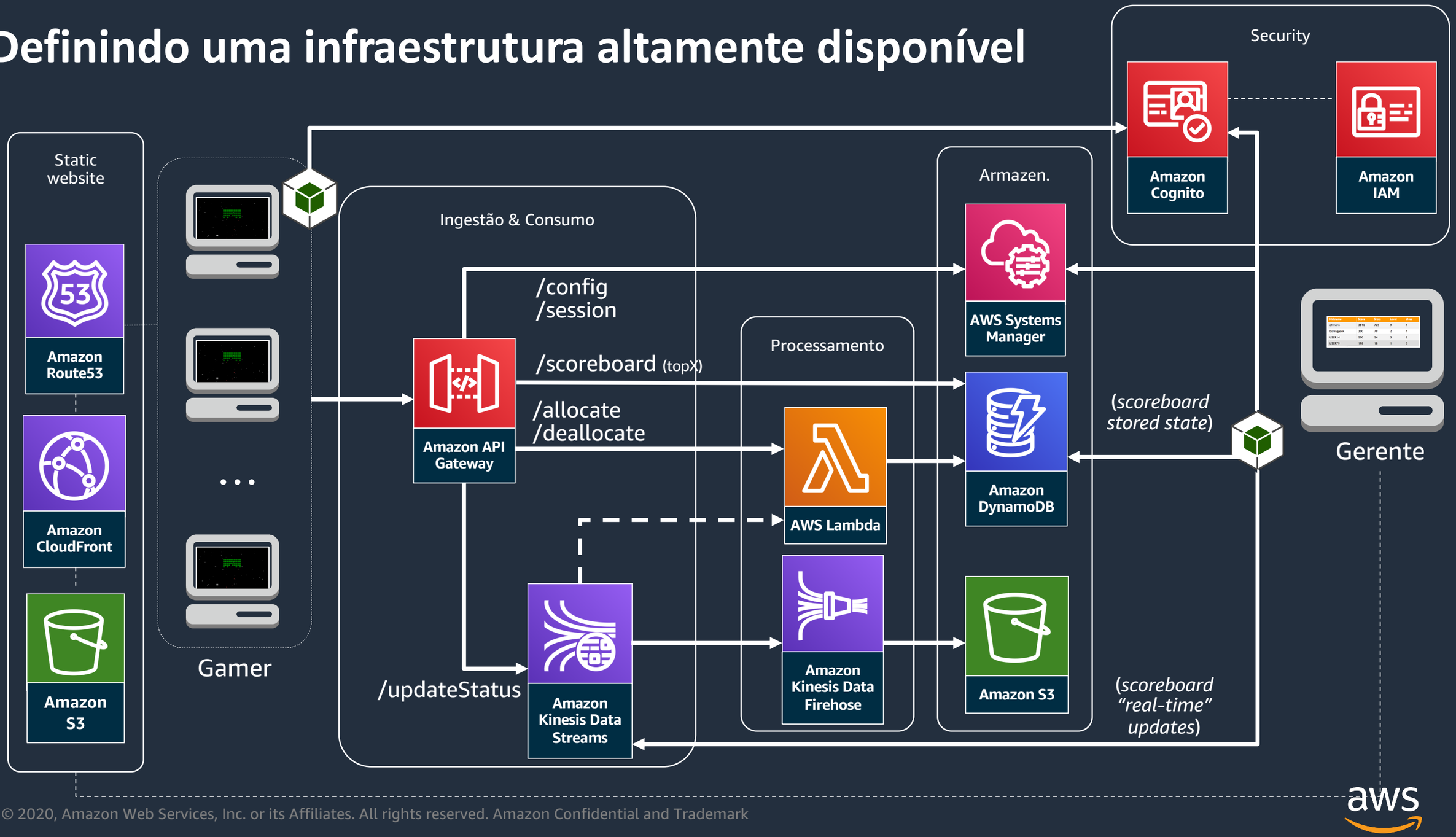


A arquitetura implementada



Revisando a arquitetura passo-a-passo

Definindo uma infraestrutura altamente disponível

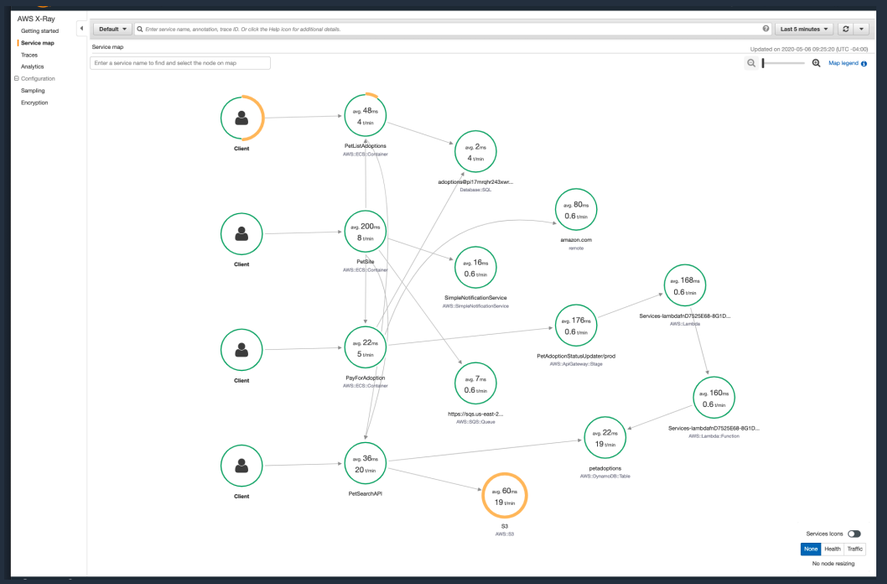


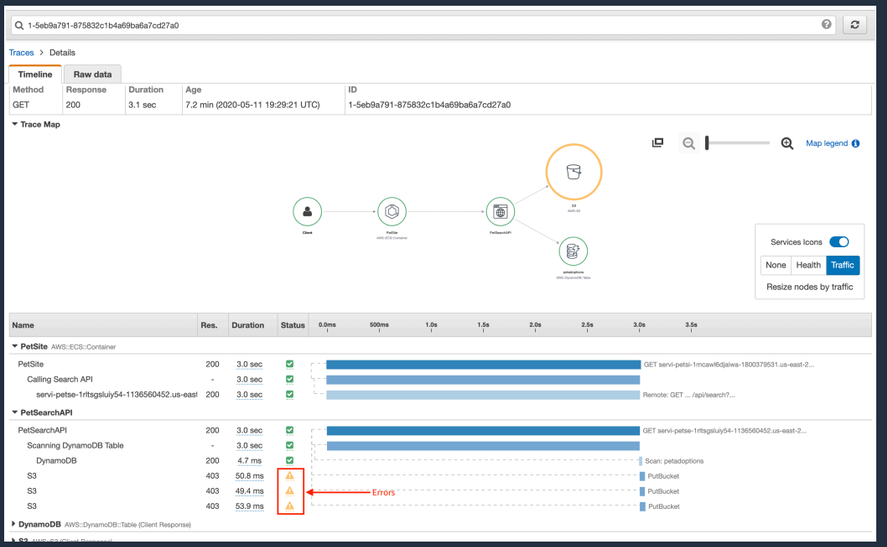
Operação

Responsabilidades da Operação

- Planejamento de Capacidade
- Updating/Upgrading/Patching
- Gerenciar Licenças
- Help/Service Desk
  - Atender a chamados de suporte (end-user e back-office)
    - Break&fix
  - Atender a pedidos de mudanças de configuração
    - Capacity update
- ...

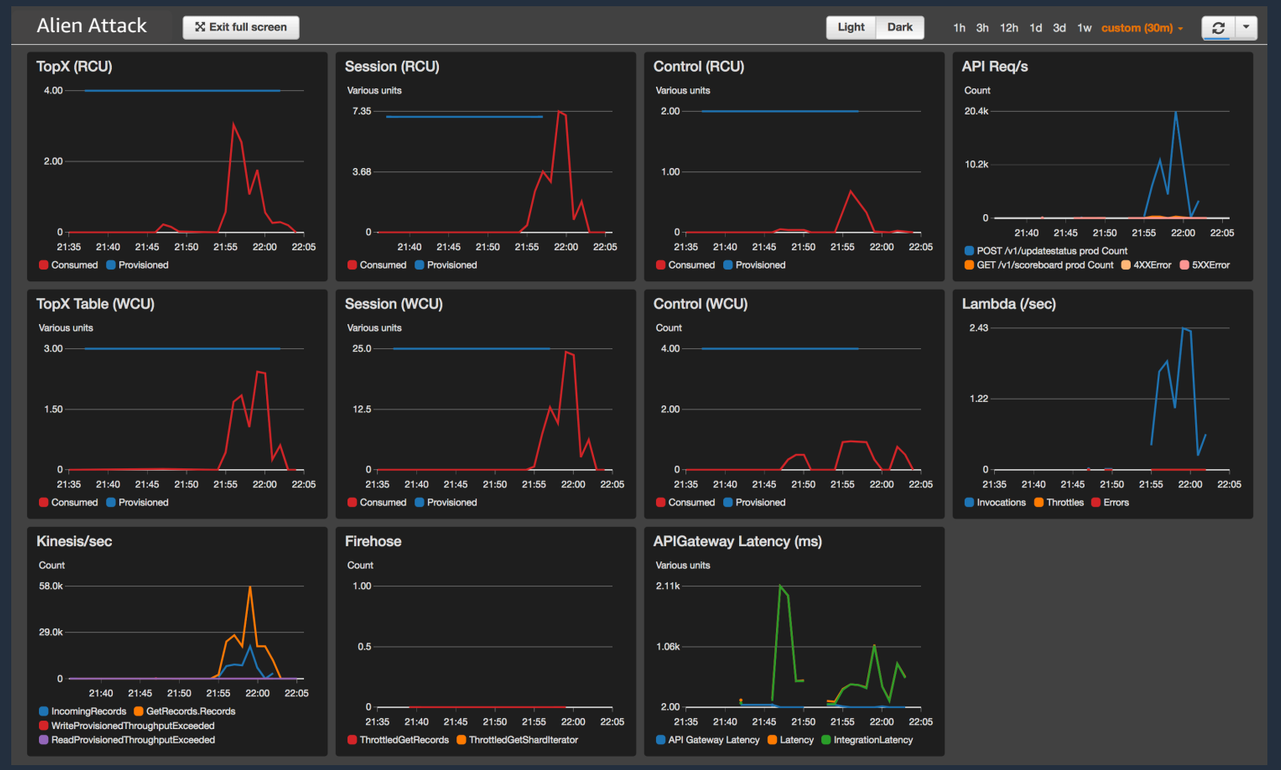
Observability



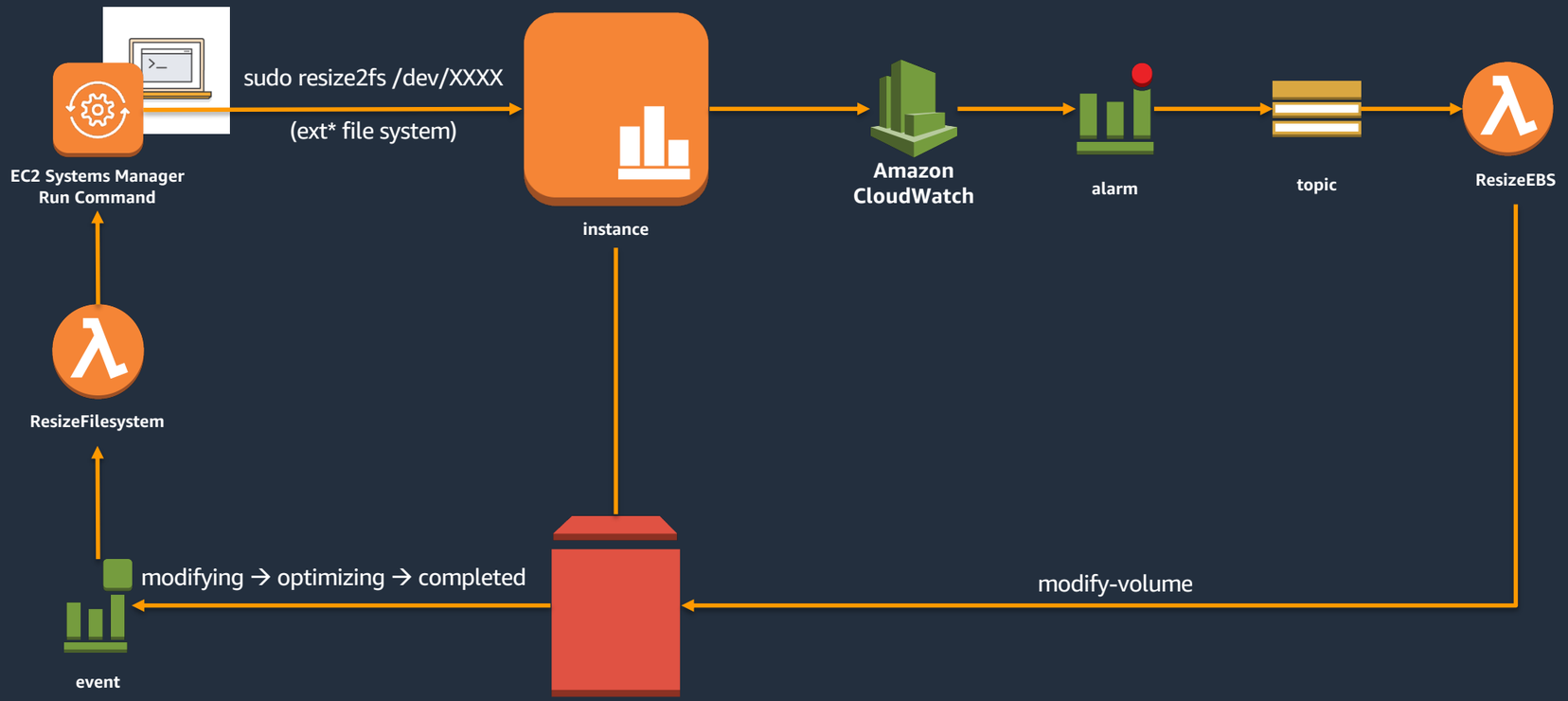


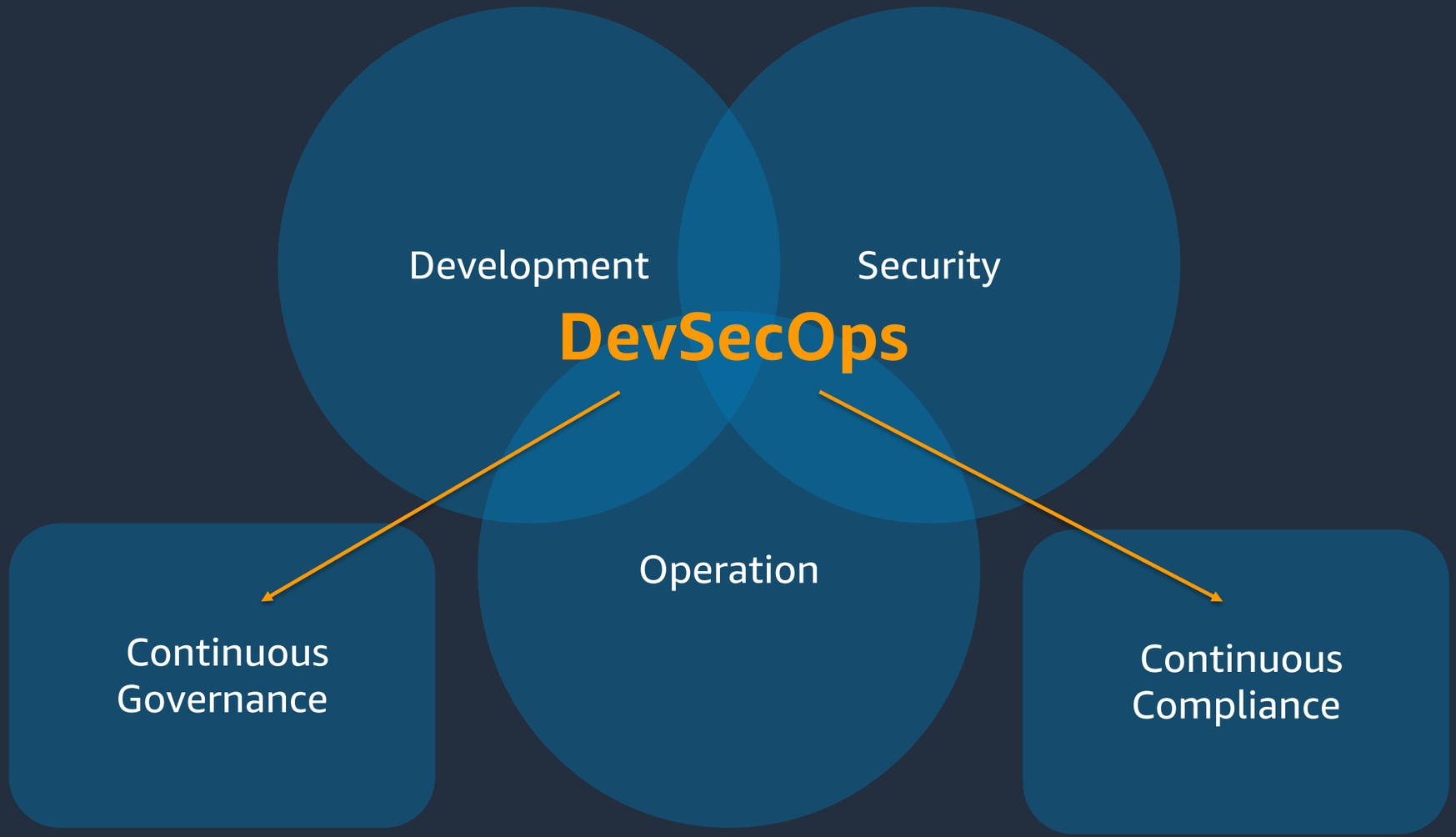
AWS X-Ray

Amazon CloudWatch



Operação reativa para Operação Proativa



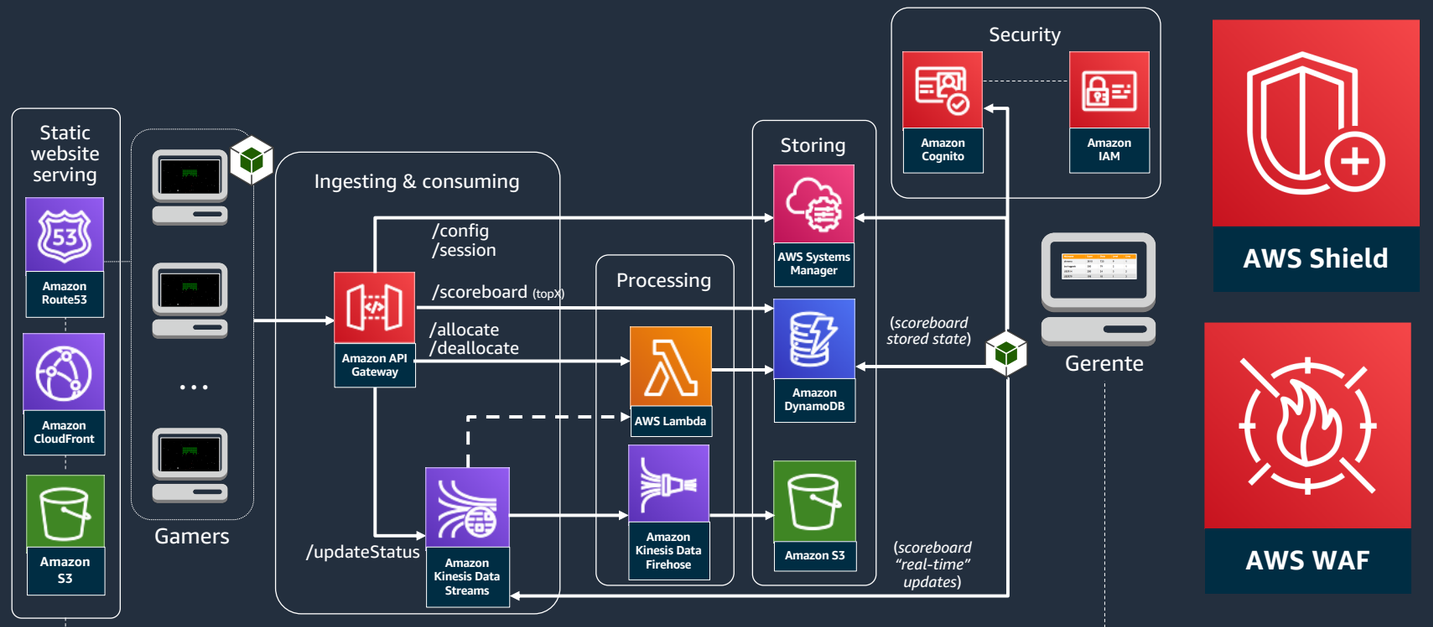


Teams of 8 to 10

- Applications
- Infrastructure
- Leadership
- Security
- Operations

Achieve speed and agility with **two-pizza teams**, while retaining reporting structures

E a partir daqui?



AWS Shield

AWS WAF

AWS Device Farm

AWS AppSync

AWS Step Functions

Amazon Simples Queue Service

AWS IoT Core

AWS Glue

Amazon Athena

Amazon QuicksSight

Amazon Sagemaker

… e muito mais

Clientes usando e construindo soluções Serverless

StreamAlert by Airbnb

StreamAlert is a serverless, real-time data analysis framework which empowers you to ingest, analyze, and alert on data from different sources and alerting logic you define.

Visit StreamAlert GitHub >>
Visit Airbnb.io >>

Cloud Custodian by Capital One

Cloud Custodian is a rules engine for managing public cloud accounts and resources. It allows users to define policies to enable a well managed cloud infrastructure and consolidates adhoc scripts into a lightweight and flexible tool.

Visit Cloud Custodian GitHub >>

Odin & Fenrir by Coinbase

Odin allows you to deploy 12-factor applications to AWS. Fenrir is a secure AWS SAM deployer that helps manage serverless projects and scale serverless across your teams and orgs.

Visit Odin GitHub >>
Visit Fenrir GitHub >>

ShadowReader by Edmunds

ShadowReader is a load-testing tool that replays production traffic to a destination of your choice by collecting traffic patterns from access logs.

Visit ShadowReader GitHub >>

Bender by Nextdoor

Bender provides an extendable Java framework for creating serverless ETL functions on AWS Lambda. Bender provides the interfaces necessary to build modules for all aspects of the ETL process.

Visit Bender GitHub >>

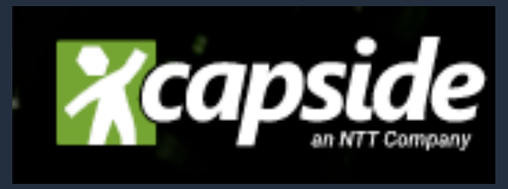
Jazz by T-Mobile

Jazz is a serverless development platform that accelerates the adoption of serverless technology within your enterprise. Jazz can help build functions, APIs, and static websites, and comes with CI/CD by default.

Visit Jazz GitHub >>

aws

Partners no Brasil



aws

#GoBuild!