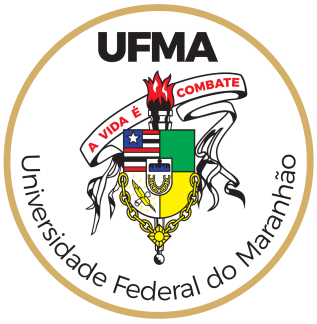


UNIVERSIDADE FEDERAL DO MARANHÃO
Curso de Graduação em Ciência da Computação

Marco Antonio Silva Pegado

Aplicação da Abordagem Serverless no Desenvolvimento de Aplicações na Nuvem: um estudo de caso com a solução AWS Lambda

São Luís - MA
2021

Marco Antonio Silva Pegado

Aplicação da Abordagem Serverless no Desenvolvimento de Aplicações na Nuvem: um estudo de caso com a solução AWS Lambda

Monografia apresentada ao curso de Ciência da Computação da Universidade Federal do Maranhão, como parte dos requisitos necessários para obtenção do grau de Bacharel em Ciência da Computação.

Curso de Graduação em Ciência da Computação

Universidade Federal do Maranhão

Orientador: Prof. Dr. Mário Antonio Meireles Teixeira

São Luís - MA

2021

Ficha gerada por meio do SIGAA/Biblioteca com dados fornecidos pelo(a) autor(a).
Diretoria Integrada de Bibliotecas/UFMA

Silva Pegado, Marco Antonio.
Aplicação da Abordagem Serverless no Desenvolvimento de Aplicações na Nuvem: um estudo de caso com a solução AWS Lambda / Marco Antonio Silva Pegado. - 2021.
55 f.

Orientador(a): Mário Antonio Meireles Teixeira.
Monografia (Graduação) - Curso de Ciência da Computação, Universidade Federal do Maranhão, UFMA, 2021.

1. AWS. 2. Eventos. 3. Nuvem. 4. Recursos computacionais. 5. Serverless. I. Meireles Teixeira, Mário Antonio. II. Título.

Marco Antonio Silva Pegado

Aplicação da Abordagem Serverless no Desenvolvimento de Aplicações na Nuvem: um estudo de caso com a solução AWS Lambda

Monografia apresentada ao curso de Ciência da Computação da Universidade Federal do Maranhão, como parte dos requisitos necessários para obtenção do grau de Bacharel em Ciência da Computação.

Trabalho de Monografia. São Luís - MA, 29 de abril de 2021:

Prof. Dr. Mário Antonio Meireles Teixeira
Orientador
Universidade Federal do Maranhão

Prof. Me. Carlos Eduardo Portela Serra de Castro
Examinador Interno
Universidade Federal do Maranhão

Prof. Me. Alana de Araujo Oliveira Meireles Teixeira
Examinador Interno
Universidade Federal do Maranhão

São Luís - MA
2021

Agradecimentos

Agradeço a Deus, pelas condições proporcionadas que me permitem correr atrás de meus objetivos.

Agradeço a meus familiares pelo apoio incondicional, pela torcida sempre forte pelo meu sucesso, e até pelos puxões de orelha, que me fazem sempre buscar corrigir minhas falhas e evoluir em todos os âmbitos da minha vida.

Pelos amigos que sempre estão ao meu lado e desejando meu sucesso. Agradeço o apoio e o companheirismo de todos.

Por fim, agradeço ao professor Mário, que me orientou no desenvolvimento deste trabalho, a professora e coordenadora Simara pela orientação fundamental neste momento final e em outros momentos, e a todo o corpo docente do curso, pelos aprendizados valiosos que serão levados por toda minha vida.

"“Os que se encantam com a prática sem a ciência são como os timoneiros que entram no navio sem timão nem bússola, nunca tendo certeza do seu destino."

(Leonardo da Vinci)

Resumo

Este trabalho propõe um estudo sobre as características e vantagens da metodologia serverless, através da análise de uma aplicação construida utilizando, entre outras ferramentas, o AWS Lambda, que é a plataforma de computação orientada a eventos e sem servidor fornecida pela Amazon. O estudo de caso se baseia em uma aplicação de reconhecimento de faces pré categorizadas, cujo funcionamento será controlado através de uma função hospedada no serviço AWS Lambda, e cuja ativação será por meio de eventos definidos previamente. Será mostrado que uma aplicação pode ser construída e estar em pleno funcionamento sem haver a necessidade de provisionamento de servidores ou instâncias virtuais, bem como será mostrado que essa metodologia pode se mostrar inclusive vantajosa à nivel financeiro em relação à outras abordagens tecnológicas.

Palavras-chave: serverless, nuvem, lambda, recursos computacionais, eventos, AWS.

Abstract

This work proposes a study on the characteristics and advantages of the methodology serverless, through the analysis of an application built using, among other things tools, AWS Lambda, which is the event-driven computing platform and without server provided by Amazon. The case study is based on an application of recognition of pre-categorized faces, whose operation will be controlled through function hosted on the AWS Lambda service, which will be activated through previously defined events. It will be shown that an application can be built and be in full operation without the need to provision servers or virtual instances, as well as it will be shown that this methodology can be shown even financially advantageous over other technological approaches.

Keywords: serverless, nuvem, lambda, recursos computacionais, eventos, AWS.

Lista de ilustrações

Figura 1 – Usos variados da computação em nuvem. 17
Figura 2 – Modelos de serviço da computação em nuvem. 19
Figura 3 – Representação dos modelos de implantação da computação em nuvem. 21
Figura 4 – Representação de regiões e zonas de disponibilidade da AWS. 24
Figura 5 – Exemplo de listagem de buckets S3 através do AWS CLI. 26
Figura 6 – O AWS Lambda é um serviço orientado a eventos. 28
Figura 7 – Exemplo de análise gráfica de uso de CPU em uma instância EC2. . . . 30
Figura 8 – Digrama de exemplo de uso do CloudWatch integrado ao EC2 e ao AWS Lambda. 31
Figura 9 – Alguns tipos de instâncias EC2 a serem escolhidas no momento de criação da instância. 32
Figura 10 – Instância criada e exibida na console do EC2, exibindo dados como status, tipo e zona de disponibilidade. 33
Figura 11 – Diagrama referente à aplicação de análise facial. 34
Figura 12 – Listagem dos buckets criados na console da AWS 35
Figura 13 – Imagens hospedadas no bucket repo-faces. 36
Figura 14 – Listagem via AWS CLI da função Lambda criada. 36
Figura 15 – SDK boto3 e alguns dos serviços acessíveis por ele. 38
Figura 16 – Código Python referente à função index_faces, no manual do SDK boto3. 39
Figura 17 – Upload das imagens selecionadas no bucket repo-faces. 39
Figura 18 – Criação da coleção via AWS CLI. 40
Figura 19 – Código usado para acessar as imagens do bucket S3. 40
Figura 20 – Imagens já indexadas à coleção. 41
Figura 21 – Imagens carregadas no bucket S3 e utilizadas na indexação. 41
Figura 22 – Configurações básicas da função Lambda. 42
Figura 23 – Evento de ativação da função Lambda configurado. 42
Figura 24 – Upload de uma foto de teste no bucket S3 via AWS CLI. 42
Figura 25 – Upload de foto para fins de teste via console da AWS. 43
Figura 26 – Exemplo de resultado da análise. 44
Figura 27 – Exemplo de resultado da análise. 45
Figura 28 – Fotos utilizadas como teste para obtenção dos resultados anteriores. . . 45
Figura 29 – Listagem de Logs de execução do Lambda no CloudWatch. 46
Figura 30 – Detalhamento de um dos logs de execução da função Lambda. 46
Figura 31 – Logs de execução visualizados dentro da console do AWS, informando duração das execuções e memória utilizada em cada uma delas. 46
Figura 32 – Gráfico referente ao número de invocações da função. 47

Figura 33 – Informações de execução da função visualizadas nas métricas do CloudWatch, informando a duração máxima e mínima. 47
Figura 34 – Informações sobre tempo de execução visualizadas num painel do CloudWatch. 48
Figura 35 – Informações visualizadas no console da ferramenta Lambda Insights. . 48
Figura 36 – Resumo dos rastreamentos de execução da função analiseFacial. 49
Figura 37 – Diagrama mostrando tempo de execução e de resposta referente a uma invocação. 49
Figura 38 – Chamada de função Lambda detalhada entre tempo de execução e de invocação. 49
Figura 39 – Configuração dos parâmetros para análise de custo do AWS Lambda. . 50
Figura 40 – Resultado da análise de custos para os parâmetros definidos anteriormente. 50
Figura 41 – Configurações para simulação de preço do Amazon EC2. 51
Figura 42 – Resultado da simulação de preços referente ao Amazon EC2. 51

Lista de abreviaturas e siglas

ACL	*Access Control List*
AMI	*Amazon Machine Image*
API	*Application Programming Interface*
AWS	*Amazon Web Services*
BaaS	*Back-end as a Service*
CLI	*Command Line Interface*
CPU	*Central Process Unit*
CSA	*Cloud Security Alliance*
EC2	*Elastic Compute Cloud*
ELB	*Elastic Load Balancing*
Faas	*Function as a Service*
IaaS	*Infrastructure as a Service*
IAM	*Identity and Access Management*
NIST	*National Institute of Standards and Technology*
PaaS	*Platform as a Service*
RDS	*Relational Database Service*
S3	*Simple Storage Service*
SaaS	*Software as a Service*
SDK	*Software Development Kit*
SNS	*Simple Notification Service*
SSH	*Secure Shell*

Sumário

1 INTRODUÇÃO . . . 12
1.1 Relevância do Tema . . . 12
1.2 Objetivo Geral . . . 12
1.3 Objetivos Específicos . . . 12
1.4 Organização do Trabalho . . . 13

2 COMPUTAÇÃO EM NUVEM . . . 14
2.1 Definição . . . 14
2.2 Características da Computação em Nuvem . . . 14
2.3 Vantagens da Computação em Nuvem . . . 15
2.4 Modelos de Serviço . . . 17
2.4.1 IaaS (Infrastructure as a Service) . . . 17
2.4.2 SaaS (Software as a Service) . . . 18
2.4.3 PaaS (Plataform as a Service) . . . 18
2.5 Modelos de Implantação . . . 19
2.5.1 Nuvem Privada . . . 19
2.5.2 Nuvem Pública . . . 20
2.5.3 Nuvem Híbrida . . . 20
2.6 Arquitetura Serverless . . . 21

3 AWS – AMAZON WEB SERVICES . . . 23
3.1 Introdução . . . 23
3.2 Amazon S3 . . . 24
3.3 AWS CLI . . . 25
3.4 AWS Lambda . . . 26
3.5 Amazon CloudWatch . . . 28
3.6 Amazon Elastic Compute Cloud (Amazon EC2) . . . 30

4 ESTUDO DE CASO: APLICAÇÃO DE RECONHECIMENTO FACIAL UTILIZANDO METODOLOGIA SERVERLESS . . . 34
4.1 Arquitetura da Aplicação . . . 34
4.2 Amazon Rekognition . . . 36
4.3 Funcionamento da Aplicação em Estudo . . . 38

5 RESULTADOS . . . 44

6 CONCLUSÃO E TRABALHOS FUTUROS . . . 52

REFERÊNCIAS 53

1 INTRODUÇÃO

1.1 Relevância do Tema

No mundo da computação, muito se tem a idéia clássica de servidores como uma regra praticamente imutável para a criação e a manutenção de soluções tecnológicas. De fato, são peças-chave no mundo da informática, e estão presentes na maioria dos serviços utilizados atualmente. Porém, em uma área tão efervescente e que se reinventa a cada dia, não é surpresa que novas abordagens e metodologias venham a surgir constantemente, e que venham a se mostrar como alternativas viáveis aos padrões e soluções já estabelecidos no mercado. A arquitetura Serverless é um exemplo.

Aparecendo como alternativa à metodologia tradicional de uso de servidores físicos, e à computação em nuvem tradicional, que faz uso de servidores virtuais criados sobre uma infraestrutura gerenciada por um provedor em nuvem, o Serverless é uma tecnologia que se propõe à eliminar a preocupação dos desenvolvedores sobre os recursos computacionais utilizados para a execução de suas aplicações, ficando toda essa responsabilidade à cargo do provedor de serviços. Vale ressaltar que diferentemente de soluções tradicionais, as aplicações baseadas em uma arquitetura Serverless são orientadas a eventos, ou seja, só haverá execução de determinada função quando ocorrer um determinado evento pré estabelecido.

Especialistas afirmam que houve um grande aumento no interesse e na utilização de soluções baseadas em Serverless por parte de diversas empresas ao redor do mundo nos últimos anos, mas também cravam que o grande salto no uso dessa metodologia ainda irá acontecer, provavelmente nesta década que se inicia, ou seja, ainda estaríamos no início desta grande mudança. Vale dizer que grandes empresas atualmente já fazem uso de soluções baseadas em Serverless, tais como Netflix, Nubank e Thomson Reuters.

1.2 Objetivo Geral

O presente trabalho tem como objetivo geral efetuar um estudo sobre a metodologia serverless computing.

1.3 Objetivos Específicos

O presente trabalho tem como principais objetivos específicos:

- Introduzir uma visão geral sobre a abordagem serverless computing, abordando seu conceito, características, funcionamento e particularidades;
- Demonstrar a eficiência da ferramenta AWS Lambda na construção de aplicações baseadas na abordagem serverless;
- Mostrar através da análise dos resultados obtidos pelo estudo de caso que as soluções baseadas na metodologia serverless podem trazem vantagens econômicas em comparação a outros tipos de abordagem computacional.

1.4 Organização do Trabalho

O Capítulo 2 traz uma introdução sobre o conceito de computação em nuvem, suas características, vantagens, modelos de implantação e de serviço, e aborda a definição e características da metodologia Serverless;

O Capítulo 3 apresenta a AWS(Amazon Web Services), plataforma voltada à computação em nuvem fornecida pela Amazon, e aborda suas principais ferramentas, entre elas as ferramentas usadas na construção da aplicação que serviu como estudo de caso, como Amazon S3 e AWS Lambda;

O Capítulo 4 mostra a construção, arquitetura e o funcionamento da aplicação usada como estudo de caso;

O Capítulo 5 apresenta os resultados apresentados pela execução da aplicação mediante ativação de eventos pré estabelecidos, bem como uma análise desses resultados e dos recursos computacionais utilizados, fazendo uso de ferramentas como o Amazon CloudWatch e o Amazon Lightsail;

O Capítulo 6 apresenta as conclusões obtidas a partir do trabalho feito, considerações finais e sugestão de temas futuros.

2 COMPUTAÇÃO EM NUVEM

Neste capítulo será definido o conceito de computação em nuvem. Será mostrada sua definição, suas principais características, vantagens do uso da computação em nuvem, além de listar e discorrer sobre seus principais modelos de serviço e de implantação. No final do capítulo também será abordada a arquitetura Serverless.

2.1 Definição

A computação em nuvem (cloud computing) pode ser definida como um modelo computacional baseado na entrega e no acesso de serviços e recursos computacionais de forma remota, podendo o acesso a estes recursos (infraestrutura, processamento, armazenamento etc.) ser feito através de dispositivos conectados à internet, sem haver a necessidade de um usuário se conectar a um computador pessoal ou a algum servidor local.

Ou seja, a nuvem nos fornece o acesso a diferentes recursos computacionais sem que eles estejam instalados ou hospedados em nossos dispositivos. Esse fornecimento de capacidade computacional feito pela nuvem é definido sob demanda, ou seja, de acordo com a necessidade do usuário em utilizar mais ou menos quantidade de um determinado recurso. Resumindo, a computação em nuvem é o fornecimento de serviços de computação, incluindo servidores, armazenamento, bancos de dados, rede, software, análise e inteligência, pela Internet (“a nuvem”) para oferecer inovações mais rápidas, recursos flexíveis e economias de escala Microsoft (2015).

A computação em nuvem surge como ótima alternativa ao modelo cliente-servidor tradicional, que normalmente faz surgir a necessidade de investimento em hardware e softwares em geral. Desta forma, uma empresa ou usuário qualquer que venha a fazer uso de uma solução tecnológica baseada em nuvem não precisará se preocupar com questões como segurança, atualizações, backup e armazenamento, sendo tudo isso responsabilidade do provedor de serviço.

2.2 Características da Computação em Nuvem

A computação em nuvem possui várias características consideradas essenciais Teleco (2018). São elas:

- Amplo Acesso À Rede: Os serviços da nuvem são disponibilizados na rede, sendo o acesso feito através de mecanismos padrões que promovem o uso de diversos tipos de

plataforma, conforme demonstrado na Figura 1. Ou seja, independente do tipo de aparelho que se esteja usando (smartphone, notebook, tec.) e de sua localização, o usuário conseguirá acessar os recursos da nuvem, bastando, para isso, possuir acesso à Internet.

- Autoatendimento sob demanda: O usuário que utiliza os serviços da nuvem pode requerer aumento ou diminuição na quantidade de recursos computacionais utilizados, tais como armazenamento e memória, sem a necessidade de interação direta com o fornecedor do serviço.

- Elasticidade rápida: O provedor de computação em nuvem tem a capacidade de provisionar mais ou menos recursos computacionais para o usuário nos momentos necessários, e de forma praticamente instantânea. Ou seja, recursos podem ser alocados de forma automática em momentos de alta demanda em uma aplicação, da mesma forma que podem ser liberados em momentos de uso menos intenso.A elasticidade é, sem dúvida, uma das características mais marcantes da computação em nuvem, sendo um diferencial desse tipo de sistema distribuído em relação a outros, como grades computacionais e peer-to-peer (RIGHI, 2013).

- Agrupamento de recursos: A nuvem permite o compartilhamento de seus serviços e recursos, de modo que eles sejam acessados de forma simultânea por vários clientes, sendo os recursos físicos e virtuais distribuídos e alocados dinamicamente. Um usuário específico não possui informações mais detalhadas sobre a real localização dos recursos que faz uso em determinado momento.

- Serviço mensurável: O serviço de computação em nuvem tem a capacidade de monitorar e controlar os recursos consumidos pelo usuário tais como armazenamento, processamento e número de máquinas virtuais, bem como de reportar essas informações ao mesmo. Isso possibilita ao cliente em posse dessas informações otimizar ao máximo a utilização desses recursos de acordo com sua real demanda, bem com possibilita ao provedor fazer a cobrança pelos recursos de forma mais justa possível.

2.3 Vantagens da Computação em Nuvem

O uso da computação em nuvem tem algumas vantagens, principalmente se comparado aos modelos tradicionais de computação. Dentre elas, destacamos redução de custos, alta disponibilidade, escalabilidade e segurança, a serem detalhados a seguir.

- Alta Disponibilidade: os provedores de computação em nuvem buscam garantir que os recursos (arquivos, aplicações etc.) estejam disponíveis em qualquer momento para serem acessados por possíveis usuários. Os serviços em nuvem oferecem a

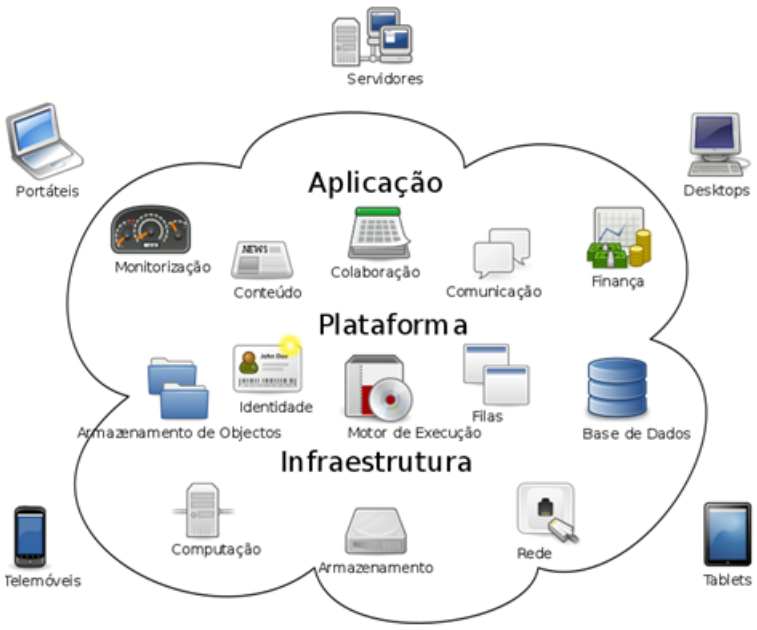
possibilidade de serem executados procedimentos preditivos (monitoramento de infraestrutura, visando impedir a ocorrência de incidentes indesejados) e corretivos (rápida identificação de paralisações e rápida solução do problema) que visam eliminar quaisquer pontos de falha. A nuvem também trabalha com redundância de recursos. Vários fatores podem comprometer e interromper o funcionamento de um determinado serviço, tais como número muito alto de requisições simultâneas e incidentes de segurança. O provedor de nuvem garante soluções eficazes para lidar com todos esses pontos, garantindo assim pleno funcionamento.

- Redução de Custos: O uso da computação em nuvem pode significar economia financeira para as empresas e organizações. Utilizando a nuvem, não há necessidade de arcar com custos referentes à hardware, licenciamento de softwares e espaço físico, facilitando principalmente os investimentos iniciais de uma empresa ou de um projeto qualquer. Além disso, utilizando a nuvem temos maior capacidade de impedir que existam recursos computacionais ociosos, garantindo assim menor gasto, pois o provedor do serviço cobrará do cliente apenas os recursos consumidos por ele.

- Segurança: Outra vantagem no uso da computação em nuvem é o seu alto grau de segurança. Os provedores oferecem vários recursos avançados de segurança, que garantem maior confiabilidade no armazenamento e manipulação de informações. Além das ações de proteção feitas pelo provedor, existem ferramentas que podem ser manipuladas pelo próprio usuário de modo a aplicar medidas próprias de segurança, garantindo ainda mais proteção.

- Escalabilidade: Ao utilizar os serviços da nuvem, o usuário terá a capacidade de aumentar ou diminuir a quantidade de recursos computacionais utilizados de forma inteligente, de acordo com a necessidade. Ou seja, sempre que houver aumento ou diminuição na demanda de usuários, os recursos computacionais (armazenamento, memória, processamento)podem ser ajustados imediatamente, visando garantir disponibilidade do serviço aos usuários externos, bem como possibilita redução de custos por parte das empresas, pois além de evitar a disponibilidade de recursos ociosos para o usuário, garante que um determinado serviço seja acessado por um número maior de usuários sem a necessidade de adquirir equipamentos físicos. O termo “elasticidade” é confundido com “escalabilidade”. A elasticidade diz respeito à capacidade, proativa ou reativa, de aumentar ou diminuir os recursos de um serviço em tempo de execução. A noção de tempo na elasticidade é crucial, envolvendo tanto o atraso para a percepção da necessidade de reconfiguração quanto a duração desse procedimento. Já a escalabilidade define a habilidade de um sistema de lidar comuma quantidade maior de carga à medida que novos recursos são adicionados, mantendo um nível de desempenho uniforme ou aproximado. Diferentemente da elasticidade, o conceito de escalabilidade é livre da noção de tempo (RIGHI, 2013).

Figura 1 – Usos variados da computação em nuvem.



Fonte: Wikipedia (2021)

2.4 Modelos de Serviço

De acordo com a definição do NIST – Instituto Nacional de Padrões e Tecnologia do Departamento de Comércio norte americano – a computação em nuvem é dividida em 3 principais modelos de serviço: SaaS (Software as a Service), IaaS (Infrastructure as a Service) e PaaS (Plataform as a Service), conforme demonstra a Figura 2. Esses modelos são importantes, pois eles definem um padrão da arquitetura para soluções de Computação em Nuvem (LIMA, 2014).

2.4.1 IaaS (Infrastructure as a Service)

Segundo definição da CSA (Cloud Security Alliance), organização sem fins lucrativos que tem como missão promover o uso das melhores práticas em computação em nuvem, a IaaS "Entrega uma infraestrutura computacional (tipicamente uma plataforma de virtualização) como um serviço, junto com armazenamento e rede brutos. Assim, em vez de comprar servidores, sistemas, armazenamento, ou equipamentos de rede, o cliente adquire tais recursos como um serviço inteiramente terceirizado." Ou seja, este modelo de serviço nos garante a disponibilidade de servidores, data centers, e recursos computacionais como processamento, armazenamento e redes. O usuário tem o poder de controle sobre pontos importantes, tais como sistemas operacionais, aplicações e espaço disponível para armazenamento, mesmo com a infraestrutura da nuvem não visível para ele. Este modelo trata do fornecimento dos recursos (processamento,capacidade de armazenamento, entre

outros) em sua forma fundamental, através da abstração em máquinas virtuais (DAROLT FELIPE RODRIGO DE SOUZA, 2016).

A IaaS é um modelo que atualmente se mostra extremamente útil e vantajoso para empresas, organizações e profissionais de tecnologia em geral. A grande vantagem deste modelo é a capacidade indefinida de provisionamento de recursos computacionais ao cliente, de modo a garantir o atendimento de suas necessidades. Ou seja, o modelo IaaS garante escalabilidade ao usuário, existindo sempre a possibilidade de fornecer mais servidores virtuais, armazenamento, processamento, entre outros recursos, garantindo assim estabilidade aos serviços e aplicações em geral.

2.4.2 SaaS (Software as a Service)

SaaS é um modelo de serviço na nuvem que possibilita a disponibilização de softwares para uso geral através da internet. A execução desses aplicativos é feita integralmente na estrutura do provedor de serviços da nuvem, que também se torna responsável por conectividade, acesso e segurança. O uso de recursos por uma determinada aplicação é escalável, ou seja, é capaz de aumentar ou diminuir de acordo com a demanda. Além disso, esses aplicativos são acessíveis a qualquer tipo de dispositivo, de qualquer localização do mundo, bastando apenas possuir acesso à internet. Neste modelo de serviço, aplicações são executadas e hospedadas em uma Nuvem de uma organização. Estas aplicações são pagas por tempo de uso (pay-as-you-go) e não é necessário aquisição de uma cópia de uso (BACELAR P. POPIOLEK, 2012).

Vários serviços famosos utilizam o modelo SaaS. Entre eles podemos citar Dropbox, Google Drive, Netflix, Paypal, entre outros. Uma das vantagens do modelo SaaS é a redução de custos, uma vez que o usuário paga apenas uma assinatura mensal, ou pelos serviços utilizados. Além disso, com o produto hospedado na nuvem, a necessidade de efetuar gastos relativos à infraestrutura diminuem consideravelmente, tornando desnecessário o investimento em equipamentos, servidores e espaço físico.

2.4.3 PaaS (Plataform as a Service)

Este modelo de serviço oferece ao usuário uma plataforma de desenvolvimento completa e remotamente hospedada na Nuvem. Este modelo possibilita às empresas e usuários em geral uma plataforma completa, que permitirá desenvolvimento, gerenciamento de hospedagem de softwares. Este modelo de serviço surge como alternativa ao paradigma tradicional, em que determinados programas são instalados e executados na máquina física de um usuário, ficando assim refém de limitações de hardware e software daquela máquina em questão, tais como sistema operacional, recursos de memória e processamento computacional (BACELAR P. POPIOLEK, 2012). Esse modelo garante ao desenvolvedor a

dedicação exclusiva ao produto a ser desenvolvido e gerenciado, delegando responsabilidades como licenças de uso e manutenção de infraestrutura à cargo do provedor do serviço em nuvem. Ele fornece sistemas operacionais desejados, ambientes de execução, SGBDs (Sistemas de Gerenciamento de Banco de Dados), softwares servidor web, entre outros, além de garantir escalonamento automático de recursos computacionais, visando sempre atender diversos requisitos técnicos apresentador pelas aplicações a serem desenvolvidas e gerenciadas pelos usuários do dos serviços PaaS.

Figura 2 – Modelos de serviço da computação em nuvem.

Fonte: Integrate (2016)

2.5 Modelos de Implantação

Podemos caracterizar o Cloud Computing como um ambiente que oferece aos seus usuários provisionamento dinâmico de recursos sob demanda em escalabilidade. A forma de cobrança se baseia no uso do recurso contratado ao invés de uma taxa fixa; o gerenciamento é centralizado e a distribuição geográfica do recurso é demonstrada de forma transparente. Esta tecnologia pode ser utilizada seguindo alguns modelos de implantação mais comuns atualmente, tais como: (SILVA, 2017)

2.5.1 Nuvem Privada

São aquelas construídas exclusivamente para um único usuário (uma empresa, por exemplo). Diferentemente de um data center privado virtual, a infraestrutura utilizada pertence ao usuário, e, portanto, ele possui total controle sobre como as aplicações são

implementadas na nuvem. Uma nuvem privada é, em geral, construída sobre um data center privado (SILVA, 2017).

Neste modelo de implantação, normalmente temos recursos computacionais diversos que serão acessados por uma só empresa, equipe ou organização. A nuvem privada pode estar disponibilizada por um provedor de serviços terceirizado, ou mesmo estar localizada fisicamente no datacenter da organização. Diferentemente de um data center virtual privado, no caso da nuvem privada temos uma infraestrutura pertencente ao próprio usuário, que por sua vez, terá total controle sobre os procedimentos adotados na nuvem. Como vantagens do uso de nuvens privadas, podemos citar alto nível de flexibilidade, maior grau de escalabilidade em comparação à infraestruturas locais, e maior nível de controle e privacidade, uma vez que não temos compartilhamento de recursos e informações com outros usuários.

2.5.2 Nuvem Pública

As nuvens públicas são aquelas que são executadas por terceiros. As aplicações de diversos usuários ficam misturadas nos sistemas de armazenamento, o que pode parecer ineficiente a princípio. Porém, se a implementação de uma nuvem pública considera questões fundamentais, como desempenho e segurança, a existência de outras aplicações sendo executadas na mesma nuvem permanece transparente tanto para os prestadores de serviços como para os usuários (SILVA, 2017).

Trata-se do modelo de implantação mais conhecido e difundido quando falamos de computação em nuvem. Nesse modelo, temos um provedor de serviços de nuvem terceirizado, que é responsável por toda a disponibilidade e gerenciamento de softwares, hardwares, procedimentos de segurança e todo e qualquer recurso computacional, que serão acessados pelos usuários através da internet. Atualmente temos vários serviços de nuvem pública, normalmente geridos por grandes empresas, tais como Azure (Microsoft) e AWS(Amazon). A utilização de modelos de nuvem pública entrega ao usuário inúmeras vantagens, como redução de custos, inexistência da necessidade de manutenção, escalabilidade e alto nível de confiabilidade.

2.5.3 Nuvem Híbrida

Além da nuvem pública e da nuvem privada, temos ainda o modelo de implantação conhecido como nuvem híbrida. Esse modelo combina o uso de nuvem privada e de nuvem pública, conforme demonstrado na Figura 3, garantindo também o tráfego de dados e aplicativos entre os dois ambientes. Esse modelo é bastante usado para garantir alta disponibilidade e desempenho aos serviços e aplicações de seus usuários. Em momentos de alta demanda, por exemplo, a o modelo de implantação híbrido permite ao usuário a

capacidade de escalar sua infraestrutura local para a nuvem. Ou seja, quando conveniente, os recursos do provedor de nuvem pública serão utilizados em conjunto com os recursos da nuvem privada, garantindo assim maior estabilidade, desempenho e disponibilidade do serviço. Também é garantido à empresa que usa este modelo a execução de cargas de trabalho específicas na nuvem pública.

Figura 3 – Representação dos modelos de implantação da computação em nuvem.

Fonte: Suite (2019)

2.6 Arquitetura Serverless

A arquitetura Serverless, também conhecida como "computação sem servidores", é uma arquitetura computacional que funciona baseada em eventos. De acordo com (AMAZON, 2020b) sem servidor é uma maneira de descrever os serviços, práticas e estratégias que permitem desenvolver aplicações mais ágeis, para que você possa inovar e responder às mudanças com mais rapidez. Através dessa arquitetura, que basicamente é um modelo de desenvolvimento nativo em nuvem, surgiu um novo paradigma em termos de desenvolvimento de software, uma vez que o serverless possibilita ao desenvolvedor criar e manter seus produtos, sem a necessidade de se preocupar com a infraestrutura e com os recursos computacionais necessários para a execução de seus programas.

O Serverless dispensa o gerenciamento de servidores para criação e execução de aplicações, mas isso não significa que na prática, eles não são utilizados. De fato, ainda são usados servidores. A diferença é que neste modelo, eles são abstraídos do desenvolvimento de aplicações. Cabe ao desenvolvedor apenas disponibilizar seus códigos fonte em containers, ficando toda a responsabilidade sobre tarefas e recursos como provisionamento, capacidade

de processamento, sistemas de armazenamento e infraestrutura em geral à cargo do provedor do serviço em nuvem.

Num modelo de nuvem baseado em IaaS (Infrastructure as a Service) o provedor de nuvem fornece ao usuário componentes de servidor "sempre ativos" para o funcionamento das aplicações. Á medida que a demanda pelo serviço aumenta ou diminui, cabe ao usuário aplicar políticas de escalabilidade, aumentando ou diminuindo os recursos destinados à hospedagem de sua aplicação. Mesmo nos casos em que o aplicativo não está em uso, a infraestrutura de nuvem alocada para a sua execução continua ativa. Nesse sentido, fica explícita a principal diferença entre o serverless e as outras soluções cloud: no modelo sem servidor, a aplicação será iniciada apenas quando necessário. Quando solicitado, um evento previamente estabelecido irá acionar a execução do código hospedado no contêiner, cabendo ao provedor de nuvem alocar dinamicamente os recursos necessários para essa execução. Os usuários serão cobrados apenas pelo tempo em que esse código estiver sendo executado.

A utilização do modelo de computação sem servidor pode garantir inúmeras vantagens ao usuário. Inicialmente, podemos citar o aumento da produtividade. Uma vez que tarefas de provisionamento e gerenciamento de recursos ficarão à cargo do provedor do serviço de nuvem, os programadores e analistas terão mais tempo para se dedicar ao desenvolvimento de suas aplicações. Com aplicativos sem servidor, o provedor de serviços de nuvem provisiona, dimensiona e gerencia automaticamente a infraestrutura necessária para executar o código. Os custos operacionais também são bem reduzidos quando usamos o serverless, uma vez que o usuário somente pagará sobre o tempo de execução de seus códigos, diferentemente de outros modelos. Atualmente, todos os grandes provedores de nuvem possuem soluções tecnológicas voltadas ao Serverless. Como exemplo podemos citar o serviço Lambda, disponibilizado pela Amazon, e o Azure Functions, da Microsoft. De acordo com (MICROSOFT, 2019) eliminando a necessidade de gerenciar a infraestrutura, a computação sem servidor permite que os desenvolvedores criem aplicativos de forma mais rápida.

As soluções de computação serverless podem ser divididas em 2 categorias: BaaS e FaaS. BaaS (Back-End as a Service) garante ao desenvolvedor o acesso à vários serviços e aplicações de back-end, facilitando assim o trabalho de programação. Como exemplos de recursos BaaS, podemos citar Bancos de Dados e serviços de autenticação. Vale ressaltar que o provisionamento e a gestão desses recursos são totalmente automáticos e invisíveis ao desenvolvedor. Já o FaaS(Function as a Service) é a categoria mais famosa do serverless, que consiste numa forma simplificada de desenvolvimento e empacotamento de códigos, a serem executados em resposta a determinados eventos pré-estabelecidos, ficando à cargo do provedor de serviços a alocação e a liberação de recursos necessários para a execução.

3 AWS – AMAZON WEB SERVICES

3.1 Introdução

A Amazon Web Services (AWS) é uma plataforma de serviços de computação em nuvem oferecida pela Amazon que provê serviços de computação, armazenamento, banco de dados, aprendizado de máquina, etc (OKITA TIAGO A. COIMBRA, 2018). A AWS é uma das principais plataformas de computação em nuvem do mundo, oferecida pela empresa Amazon. Seus serviços e recursos são utilizados por milhares de clientes, como startups, grandes empresas e órgãos governamentais. Dentre esses serviços oferecidos, podemos destacar: armazenamento, banco de dados, inteligência artificial, machine learning, entre vários outros.

O uso do AWS possui vários benefícios. Entre eles, podemos apontar o alto grau de segurança. Trata-se de um dos serviços de computação em nuvem mais seguros da atualidade. Sua infraestrutura foi construída visando proteção de informações, dispositivos e aplicações, e garante atendimento aos principais requisitos de segurança, tais como localidade, proteção e confidencialidade dos dados.

Outra vantagem da AWS é o modelo de pagamento “Pay as You Go”. Este modelo implica que o os custos do usuário são calculados apenas de acordo com os recursos individuais que venham a ser utilizados, e pelo tempo utilizado, sem haver custos adicionais quando o recurso deixa de ser utilizado. Outro aspecto importante do AWS é que ele garante elasticidade, ou seja, através dele conseguimos flexibilizar a concessão de recursos computacionais (máquinas virtuais, armazenamento, memória etc.) de acordo com as nossas demandas de utilização, modelo este que garante maior controle dos custos. A infraestrutura da AWS também fornece a seus usuários grande disponibilidade de rede. Ou seja, as aplicações e serviços estarão disponíveis pelo maior tempo possível aos usuários. A AWS possui uma infraestrutura física que não se encontra em um determinado lugar só, mas sim espalhada em várias regiões do mundo. Uma região consiste em uma estrutura da Amazon específica, isolada em um determinado local do mundo. Existem regiões AWS na américa do sul, Canadá, Leste dos Estados Unidos, entre outros locais espalhados pelo mundo, de modo que cada região é 100% independente das outras.

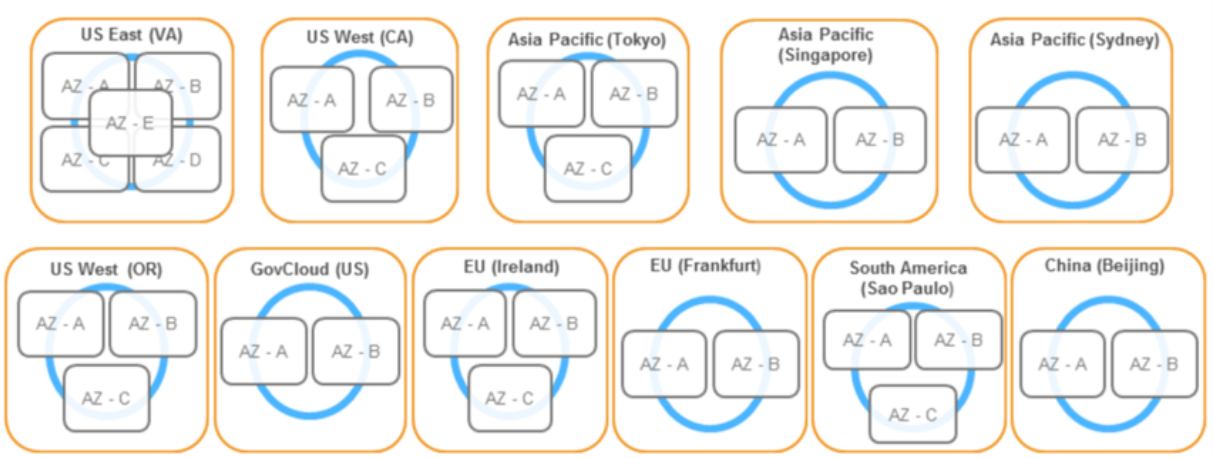
A Amazon possui datacenters em diferentes regiões no mundo, sendo que cada região é isolada geograficamente de outra, isto é, instâncias de uma região não podem se comunicar diretamente (por IP privado) com instãncias de outras regiões (OKITA TIAGO A. COIMBRA, 2018).

Cada uma dessas regiões é dividida em zonas de disponibilidade, podendo a região

ser dividida em 3, 4 ou até 6 dessas zonas, à exemplo da região do leste dos Estados Unidos. A organização da estrutura da AWS entre zonas e regiões é mostrada no esquema da Figura 4. A escolha da região pode influenciar em determinador fatores, como a latência de rede. A latência de um usuário da América do Sul que utiliza a região de São Paulo, por exemplo, será menor do que a latência existente caso esse usuário utilize uma região dos Estados Unidos ou de algum outro país mais distante.

Uma zona de disponibilidade é um conjunto de data centers, que apresenta alto grau de redundância, e trabalha com 100% de independência das outras zonas localizadas dentro da mesma região. Essas zonas de disponibilidade ficam relativamente distantes uma das outras, de modo que um determinado evento que possa vir a causar impactos em uma zona (tal como um desastre natural) tenha menor probabilidade de impactar nas outras zonas da mesma região, visando assim garantir pelo menos uma zona disponível aos usuários. As zonas de uma mesma região são conectadas entre si via redes de fibra ótica que permitem a comunicação com baixa latência, possibilitando assim as estratégias de replicação e redundância, e garantindo o alto grau de disponibilidade apresentado pela infraestrutura da AWS.

Figura 4 – Representação de regiões e zonas de disponibilidade da AWS.



Fonte: Estabil (2018)

3.2 Amazon S3

Um dos serviços oferecidos pela Amazon Web Services é o Simple Storage Service, também conhecido como Amazon S3. Trata-se de um serviço de armazenamento e recuperação de objetos, que fornece ao usuário várias vantagens tais como performance, escalabilidade e disponibilidade dos dados salvos. Uma vez disponíveis, estes dados poderão ser usados em uma infinidade de casos de uso, tais como análises de big data, dispositivos móveis, sites etc.

Para fazer este armazenamento, e disponibilizá-lo ao usuário de maneira simples e intuitiva, o S3 faz uso de uma estrutura chamada bucket S3, que são basicamente contêineres onde são depositados os objetos. Para armazenar um arquivo em um bucket, basta fazer o upload dele (via interface web ou via linha de comando), que já estará pronto para ser usado. Uma vez feito esse upload, pode-se também definir permissões ao bucket e aos seus arquivos, ou seja, determinar quem pode criar, excluir e listar objetos no bucket.

O serviço S3 é voltado para o armazenamento de objetos em buckets. Podemos armazenar quaisquer objetos com menos de 5TB e quantos objetos quisermos dentro de um bucket. Além disso, seu acesso é controlado pelo usuário criador ou administrador da conta (OKITA TIAGO A. COIMBRA, 2018).

A utilização do serviço Amazon S3 possui várias vantagens. Trata-se de um serviço altamente escalável, ou seja, é possível fazer a concessão ou retirada de recursos de armazenamento, de acordo com as demandas pontuais de cada usuário. Além disso, garante uma capacidade considerável de armazenamento de dados, haja visto que é possível carregar uma quantidade indefinida de objetos em um bucket, e que cada objeto pode vir a conter até 5 TB de tamanho. Fornece também alta disponibilidade de informações, pois replica os dados em outros servidores da infraestrutura da AWS. E seguindo os princípios básicos da computação em nuvem, o uso do S3 pode significar custos operacionais mais baixos para uma organização, já que o valor a ser pago é definido conforme a utilização, além de proporcionar uma menor necessidade de investimento em hardware.

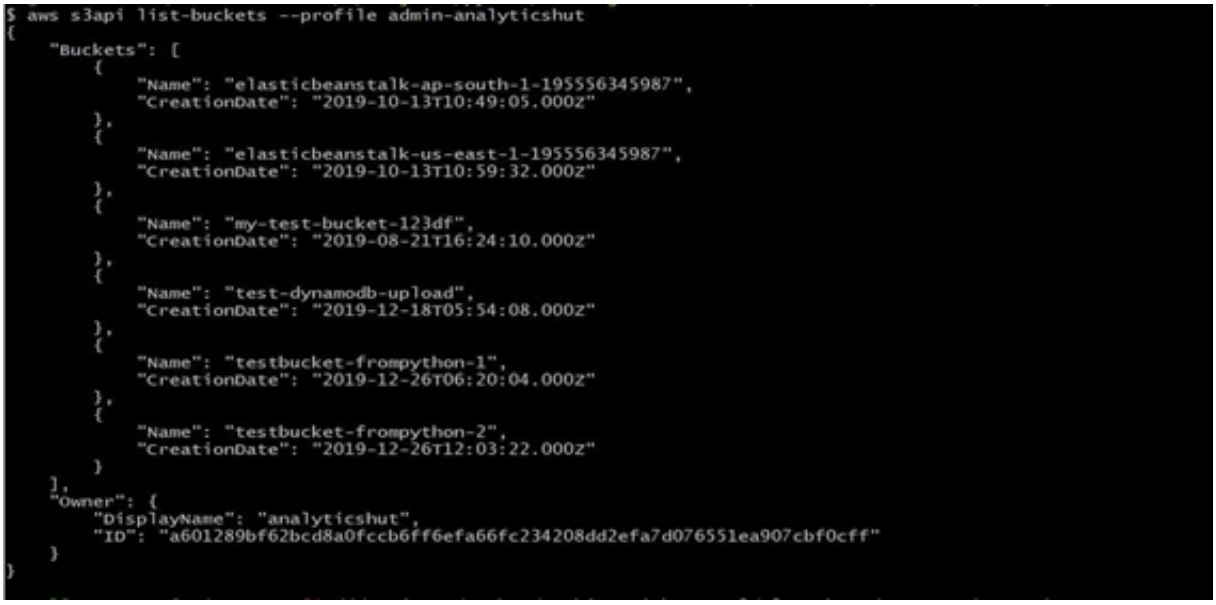
Outro fator importante a ser destacado no serviço Amazon S3 é a segurança. Por padrão, um usuário do serviço só possui acesso aos objetos inseridos por si mesmo. Se necessário, um usuário pode conceder acesso a outros usuários usando recursos de gerenciamento de acesso próprios da AWS, tais como: AWS IAM (Identity and Access Management) responsável pela criação de usuários e concessão de acessos específicos aos objeto; ACLs(Listas de Controle de Acesso) que tornam os objetos visíveis apenas para usuários específicos; e políticas de bucket, através das quais se pode configurar permissões de acesso para um bucket, bem como para todos os seus objetos. Possui o S3 Block Public Access, um recurso que garante que buckets e seus dados não tenham acesso público, e o Access Analyzer for S3, que é um serviço responsável por analisar políticas de acesso ao bucket, garantindo que as políticas forneçam apenas níveis de acesso desejados pelo usuário.

3.3 AWS CLI

Qualquer funcionalidade básica da AWS que é feita pela interface web também pode ser feita via linha de comando. Isso se torna possível com o uso da ferramenta AWS CLI (Command Line Interface). Essa ferramenta possibilita ao usuário o gerenciamento

de todos os seus serviços AWS (EC2, S3 etc.). Com ela podemos criar bases de dados, listas instâncias virtuais ativas, ter acesso aos objetos de um determinado bucket, tudo sem precisar acessar a interface da Amazon. O AWS CLI é uma ferramenta de código aberto, e está apta para funcionar em qualquer sistema operacional. O usuário da AWS CLI acessa os recursos através de duas chaves (access & secret key), sendo de fundamental importância não as compartilhar, já que um usuário não autorizado que porventura viesse a ter acesso às chaves teria acesso também à mesma gama que recursos e informações que o usuário original. Na Figura 5 temos um exemplo de uso do AWS CLI para listagem de buckets S3.

Figura 5 – Exemplo de listagem de buckets S3 através do AWS CLI.



Fonte: analyticshut (2021)

3.4 AWS Lambda

O AWS Lambda é um dos principais produtos da AWS. Trata-se de um serviço que permite ao usuário executar aplicações e procedimentos em modelo serverless, ou seja, sem a necessidade de se preocupar com servidores e infraestrutura de um modo geral. Trata-se de um tipo de serviço que possui uma tendência cada vez maior de utilização nos próximos anos, pois é um modelo de desenvolvimento em que o programador terá a liberdade de se preocupar única e exclusivamente com a implementação de seu produto, deixando toda a preocupação com administração de recursos computacionais, integridade de servidor, execução do código propriamente dita, sistema operacional, provisionamento de capacidade, escalabilidade automática e desempenho à cargo do provedor do próprio serviço Lambda. O AWS Lambda possui suporte às principais linguagens de programação

utilizadas no mercado atualmente (Java, Go, .Net etc.), podendo assim ser utilizada por boa parte dos desenvolvedores.

Lambda é um serviço da AWS que executa códigos (scripts Python ou NodeJS por exemplo) sem a necessidade de alocação de recurso computacional por parte do usuário. O serviço pode ser utilizado para execução de código em resposta a eventos, como os gerados pelo CloudWatch, por exemplo. A vantagem de utilização desses serviços é a automatização de monitoramento e respostas. Ou seja, além de monitorar recursos utilizando o CloudWatch, podemos gerar eventos para serem processados pelo Lambda automaticamente (OKITA TIAGO A. COIMBRA, 2018).

O AWS Lambda, a exemplo de outros serviços da AWS, também utiliza o modelo "pay-as-you-go". Ou seja, as despesas do usuário serão referentes apenas aos recursos computacionais utilizados, fazendo do Lambda um serviço bastante econômico em determinados casos, já que não haverá pagamento efetuado sobre recursos computacionais ociosos. Esse valor referente às despesas do uso do Lambda dependerá de fatores como quantidade de vezes em que um determinado código foi processado, tempo de execução, memória consumida etc. O AWS Lambda é um serviço de computação sem servidor que permite executar código sem provisionar ou gerenciar servidores, criando lógica de dimensionamento de cluster com reconhecimento de workloads, mantendo integrações de eventos ou gerenciando tempos de execução (AMAZON, 2014).

A inexistência de pagamentos referentes a recursos computacionais ociosos que o Lambda apresenta se deve a uma de suas principais características: em vez de disponibilidade 24/7, este serviço é orientado a eventos, conforme demonstrado no esquema da Figura 6. Ou seja, um determinado trecho de código será executado somente quando necessário, de acordo com as condições estabelecidas pelo usuário, que podem ser alterações de dados, intervenções de usuários, ou acionamento direto através de outros serviços AWS, como alterações de dados em um bucket S3, inserção de informação no Amazon DinamoDB(serviço de banco de dados NoSQL fornecido pela AWS), logs do CloudWatch, entre outros. Esse modelo se torna interessante para aplicações que são utilizadas de forma intermitente.

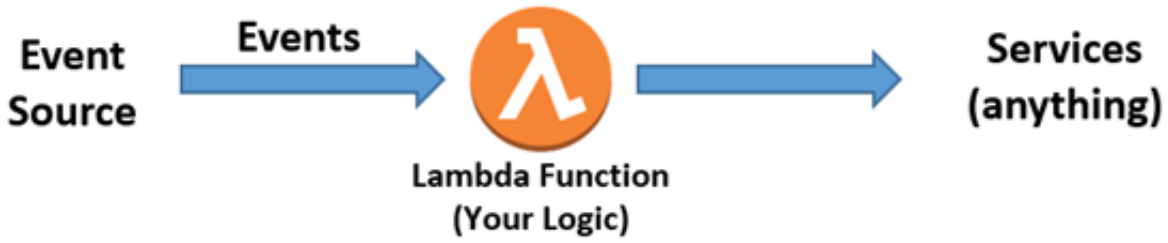
Com o AWS Lambda, você paga apenas pelo tempo de computação que consome, portanto, nunca pagará por infraestrutura superprovisionada. Você é cobrado por cada milissegundo em que seu código é executado e pelo número de vezes que ele é acionado (AMAZON, 2014).

Além da inexistência de cobrança sobre recursos ociosos, o uso do Lambda possui outras vantagens. Uma delas é a quantidade de código reduzida a ser utilizada pelo desenvolvedor em determinados casos. No Lambda teremos a execução do que realmente é necessário para a necessidade do usuário, eliminando a necessidade de ações como configuração de conexão com banco de dados, gerenciamento de rotas, entre outras configurações. Outra importante vantagem do uso deste serviço é seu alto grau de

escalabilidade. Ou seja, é possível escalar automaticamente uma função Lambda, de modo que ela suporte um número maior ou menor de requisições simultâneas, de acordo com a demanda. Todo esse processo de alocar e desalocar recursos para a função fica a cargo da AWS. Através de uma configuração chamada "Number of concurrent requests", é possível fazer este dimensionamento, que garantirá a abertura em paralelo de vários processos que executam a mesma função original.

De um modo geral, AWS Lambda se mostra um serviço eficiente, vantajoso e econômico em vários tipos de aplicações, cabendo ao desenvolvedor/analista saber identificar os melhores momentos e situações para utilizá-lo. Para casos em que o usuário precisará ter maior controle sobre seus recursos computacionais, é recomendado o uso de outros serviços nativos da AWS, tais como Amazon Elastic Compute Cloud (Amazon EC2).

Figura 6 – O AWS Lambda é um serviço orientado a eventos.



Fonte: Amazon (2020b)

3.5 Amazon CloudWatch

O Amazon CloudWatch é um serviço de monitoramento e observação criado para engenheiros de DevOps, desenvolvedores, Site Reliability Engineers (SREs – Engenheiros de confiabilidade de sites) e gerentes de TI. O CloudWatch fornece dados e insights práticos para monitorar aplicativos, responder às alterações de performance em todo o sistema, otimizar a utilização de recursos e obter uma visualização unificada da integridade operacional(AMAZON, 2019).

O Amazon CloudWatch é um serviço da AWS que possibilita ao usuário fazer o monitoramento de aplicações e recursos da AWS de acordo com as suas necessidades. Ele nos fornece métricas e logs referentes ao desempenho de variadas tarefas, procedimentos e aplicações, proporcionando assim maior poder de gerenciamento e análise ao usuário, pois além de nos fornecer informações detalhadas sobre funcionamento geral, também nos permite fazer alterações nos recursos monitorados com base em regras estabelecidas previamente, além de possibilitar o envio de notificações e a criação de alarmes, a serem

acionados quando determinadas situações estabelecidas de antemão pelo usuário ocorrerem. Através desse serviço podemos, por exemplo, monitorar o uso de CPU de uma instância EC2, de modo a possibilitar a criação automática de instâncias adicionais em momentos de cargas de trabalho mais altas, possibilitando assim o funcionamento da aplicação, ou mesmo criar alarmes que garantirão o envio de notificações em caso dessa ou de outra situação inadequada ocorrer. Da mesma forma, através do monitoramento feito pelo CloudWatch, também poderíamos automatizar a interrupção de instâncias subutilizadas em momentos de pouca demanda, garantindo assim economia financeira ao usuário em questão.

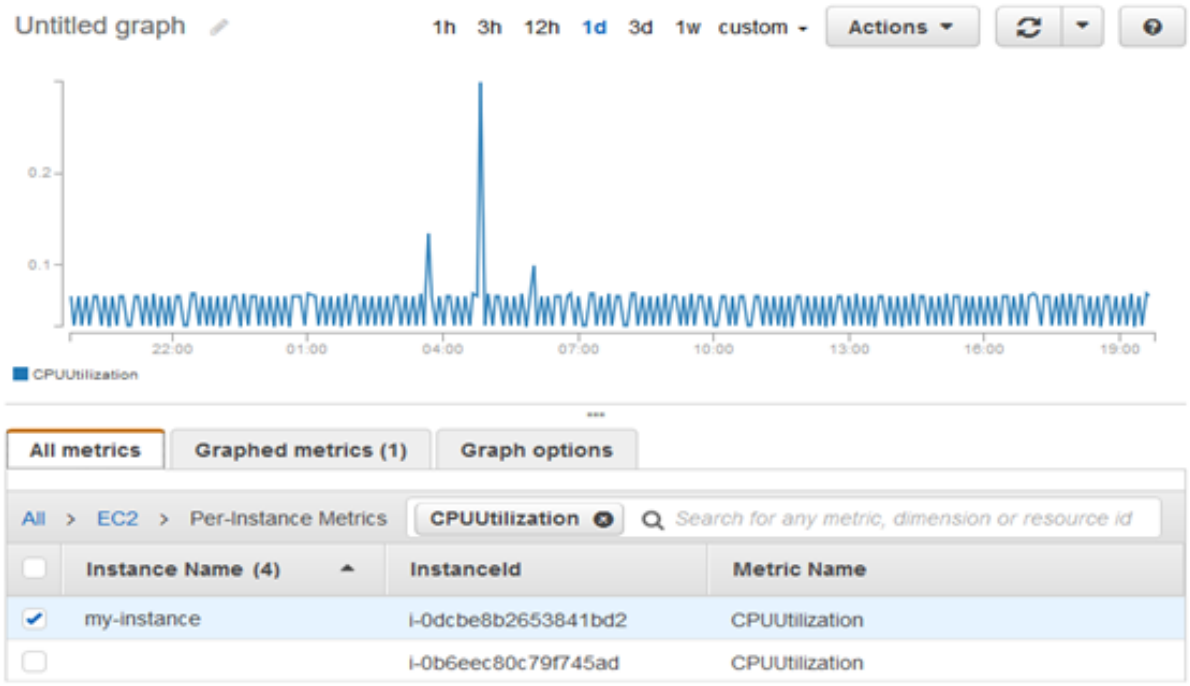
O acesso ao CloudWatch pelo usuário pode ser feito de várias formas. A principal é pelo próprio console do Amazon CloudWatch, onde é possível criar alarmes, painéis de monitoramento, visualização de logs, definição de métricas e definição de eventos de ativação, tudo de forma bem prática e intuitiva. Outra forma de acesso ao CloudWatch é através do AWS Command Line Interface (CLI), onde todas as funcionalidades citadas anteriormente também serão gerenciáveis, porém via linha comando. Existe ainda a possibilidade de usar SDKs próprios da AWS, através dos quais podemos acessar o CloudWatch usando uma API personalizada, disponível para várias linguagens de programação e plataformas.

Alguns serviços da AWS podem ser usados em conjunto com o AWS CloudWatch, conforme exemplificado na Figura 8. Um deles é o Amazon EC2 Auto Scaling: este recurso permite ao usuário promover o gerenciamento automático de instâncias EC2, de acordo com determinados requisitos e condições. Pode-se iniciar e encerrar instâncias EC2 de forma automática, tendo como base esses requisitos. Outro serviço possível de ser integrado ao CloudWatch é o SNS (Amazon Simple Notification Service), que é basicamente um serviço desenvolvido pela Amazon que visa o envio de mensagens móveis e corporativas, e permite configuração, operação e envio de notificações.

Uma vez integrado ao CloudWatch, é possível notificar determinados grupos interessados em saber que um limite de alarme foi atingido, por exemplo. O CloudWatch pode também atuar em conjunto com vários outros serviços do AWS, como AWS Lambda, Amazon S3, Amazon DynamoDB, entre outros. O CloudWatch tem seu funcionamento baseado no conceito de métricas. Uma vez que um serviço envia métricas ao CloudWatch, podemos fazer análises estatísticas dessas informações, inclusive sob a forma de gráficos de desempenho.

As métricas do AWS CloudWatch podem ser definidas como uma determinada variável ou característica a ser monitorada de acordo com a necessidade do usuário. O serviço nos oferece uma grande quantidade de variáveis possíveis a serem analisadas, tais como: utilização de CPU(conforme exemplificado na Figura 7), verificação de status de uma instância, armazenamento e número de objetos em um bucket S3 etc. A análise de informações de uma determinada métrica deve ser associada a um Time Stamp para que a

Figura 7 – Exemplo de análise gráfica de uso de CPU em uma instância EC2.



Fonte: Amazon (2020b)

análise possa ser feita. Ou seja, essa análise de informações pode ser feita com referência a um determinado espaço de tempo, que pode variar de horas até dias. A ferramenta nos oferece um ótimo grau de análise estatística das informações. Dada uma métrica, podemos saber qual o menor valor da métrica em um determinado período, o maior valor, somatório, média, entre outras. Através dos valores retornados pelas aplicações das métricas é que também poderemos fazer o acionamento dos alarmes definidos no CloudWatch. Ou seja, o alarme monitora de forma permanente os resultados de uma métrica específica, e executa uma determinada ação de forma automática, caso os resultados atinjam um valor previamente estabelecido para a ativação de um alarme.

3.6 Amazon Elastic Compute Cloud (Amazon EC2)

Um dos principais serviços da AWS, o Amazon EC2 fornece ao usuário capacidade computacional hospedada na nuvem, através da criação de servidores virtuais, que são basicamente instâncias de computadores na nuvem. O uso do EC2 permite aos profissionais maior velocidade na implantação e desenvolvimento de seus produtos, pois elimina a necessidade de investimento em hardware e infraestrutura de um modo geral, uma vez que garante o gerenciamento e a obtenção de recursos computacionais com mínimo esforço e grande facilidade. O Amazon EC2 é um serviço que apresenta elasticidade, ou seja, tem capacidade de prover mais ou menos recursos computacionais de acordo com a demanda e

Figura 8 – Digrama de exemplo de uso do CloudWatch integrado ao EC2 e ao AWS Lambda.

Fonte: Edureka (2020)

os picos de tráfego.

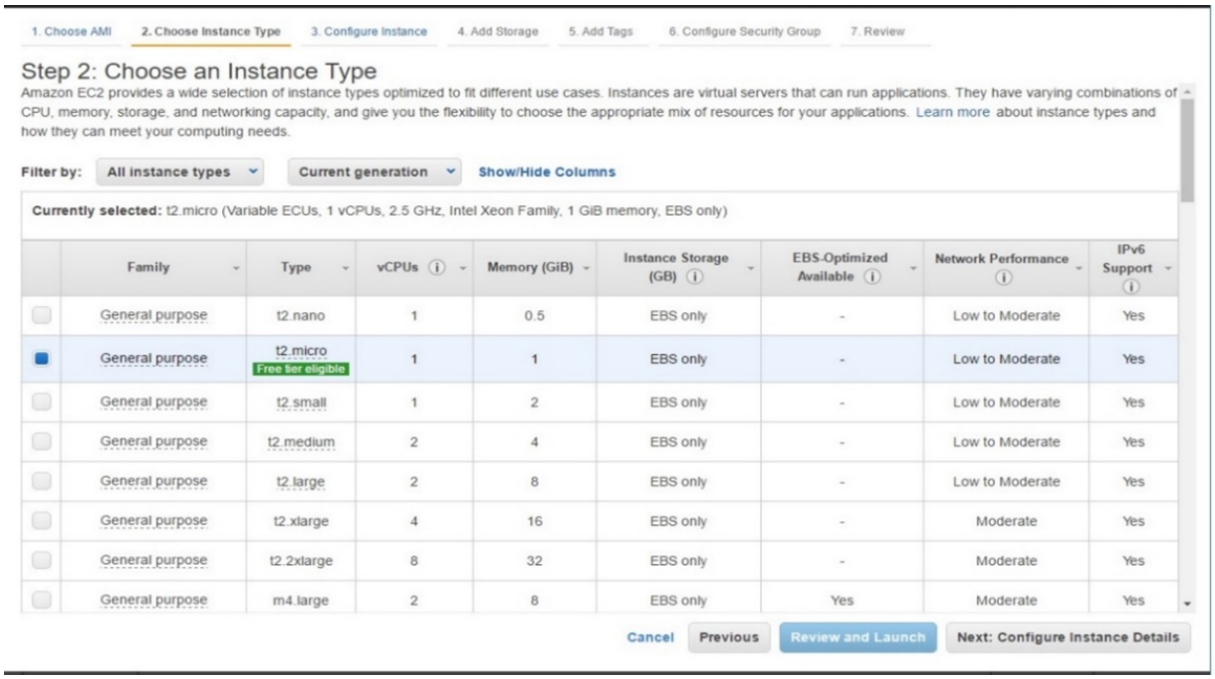
O Amazon Elastic Compute Cloud (Amazon EC2) é um serviço Web que disponibiliza capacidade computacional segura e redimensionável na nuvem. Ele foi projetado para facilitar a computação em nuvem na escala da web para os desenvolvedores. A interface de serviço Web simples do Amazon EC2 permite que você obtenha e configure a capacidade sem muito esforço. Oferece um controle completo de seus recursos computacionais e permite que você trabalhe no ambiente computacional comprovado da Amazon (AMAZON, 2020a).

O AWS EC2 fornece ao usuário a criação de ambientes de computação virtual, conhecido como instâncias. O processo de criação da instância oferece grande flexibilidade, pois podemos criar uma instância com as características necessárias ao nosso produto. O EC2 permite, por exemplo, que o usuário escolha uma AMI (Amazon Machine Image). Essa AMI contém configurações importantes como o sistema operacional escolhido (Ubuntu, Windows etc.), tipo de instância, entre outros, de forma a permitir a melhor configuração possível de acordo com as necessidades computacionais do usuário.

Os tipos de instância EC2 também são bem numerosos, de modo que cada um possua características específicas, tais como capacidade de CPU, memória e armazenamento, conforme é demonstrado na Figura 9. Cada tipo de instância é recomendado para determinados tipos de aplicações, de acordo com suas características. Ou seja, existem tipos de instâncias que terão melhor desempenho em servidores WEB, outros que serão recomendados para aplicações que envolvam aprendizado de máquina, e assim por diante. Os preços a serem cobrados, assim como em outros serviços da AWS, também são proporcionais à quantidade de recursos consumidos.

O AWS EC2 garante alto nível de disponibilidade e confiabilidade às aplicações e serviços em geral. Um dos serviços que garantem essas características é o Amazon EC2 Auto Scaling. Através do Auto Scaling, uma aplicação sempre terá disponível para si o

Figura 9 – Alguns tipos de instâncias EC2 a serem escolhidas no momento de criação da instância.



Fonte: educoutinho (2017)

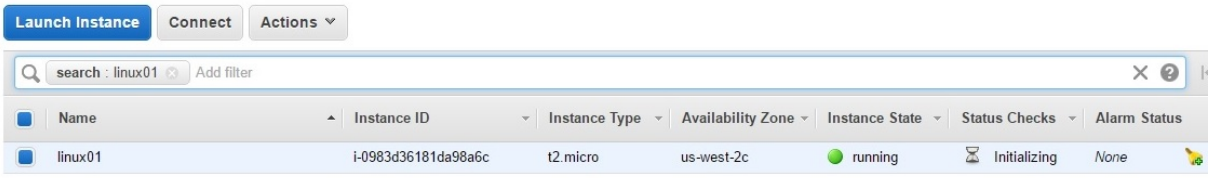
número exato de instâncias necessárias para seu funcionamento. O Auto Scaling garante a criação de novas instâncias EC2 em momentos de alta demanda, garantindo assim alto desempenho e estabilidade, assim como permite diminuir o número de instâncias ativas em momento de maior ociosidade, garantindo assim menores gastos ao usuário. Ele permite ainda que haja a substituição automática de instâncias com algum tipo de falha por novas, mais uma vez garantindo a disponibilidade do serviço.

Outro serviço interessante que auxilia o AWS EC2 é o Amazon Elastic Load Balancing (ELB). Este serviço é responsável por garantir a divisão e equilíbrio de cargas de trabalho, distribuindo o tráfego de entrada entre os demais destinos possíveis, tais como instâncias EC2. Ou seja, se porventura houver um crescimento súbito no tráfego, ou um servidor vier a ficar inativo, o ELB garantirá a redistribuição deste tráfego para os diversos servidores online. Se um novo servidor for adicionado, o Load Balancer será responsável também por iniciar automaticamente o envio de solicitações ao mesmo. Existe mais de um tipo de balanceador de carga, tais como o Application Load Balancer, o Classic Load Balancer, entre outros, cabendo ao usuário selecionar o tipo mais recomendável à sua aplicação.

O AWS EC2 pode ser acessado sob mais de uma forma. A forma mais simples é através da interface de usuário, onde é possível executar várias operações sobre as instâncias EC2 existentes, tais como interromper, encerrar e ativar a instância em questão, conforme é demonstrado na Figura 10. As instâncias também podem ser acessadas via

linha de comando, utilizando-se a ferramenta CLI, ou mesmo via um serviço SSH.

Figura 10 – Instância criada e exibida na console do EC2, exibindo dados como status, tipo e zona de disponibilidade.



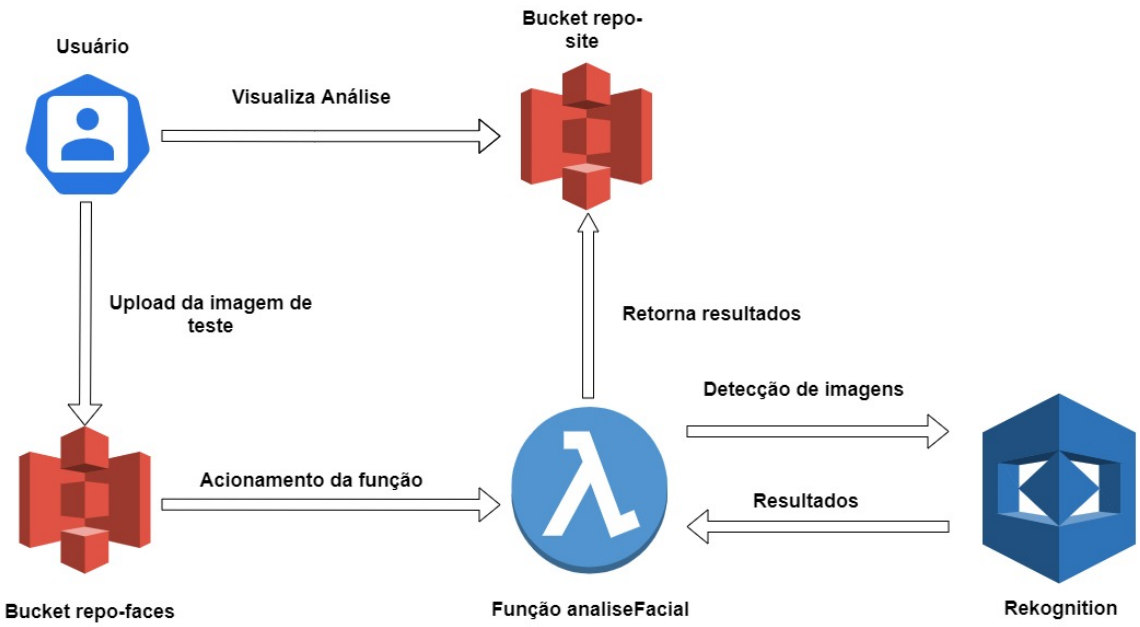
Fonte: educoutinho (2017)

4 ESTUDO DE CASO: APLICAÇÃO DE RECONHECIMENTO FACIAL UTILIZANDO METODOLOGIA SERVERLESS

Neste capítulo, visando fornecer uma perspectiva mais prática a respeito da metodologia serverless, será apresentado um estudo de caso feito sobre uma aplicação que utiliza indexação e reconhecimento de imagens, cujo funcionamento é controlado por uma função hospedada na ferramenta AWS Lambda, que é a plataforma de computação orientada a eventos e sem servidor fornecida pela Amazon. Serão apresentadas as tecnologias baseadas em nuvem utilizadas no projeto e a função de cada uma delas no escopo da aplicação, a arquitetura geral e descritos os procedimentos realizados para obtenção dos resultados desejados. Além disso, será apresentada uma visão geral a respeito do serviço AWS Rekognition e de seus principais recursos.

4.1 Arquitetura da Aplicação

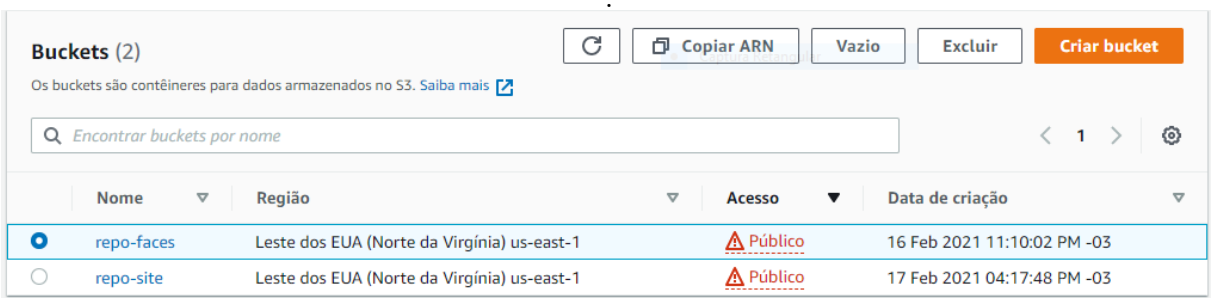
Figura 11 – Diagrama referente à aplicação de análise facial.



Fonte: Autor

Na construção da aplicação de análise facial utilizada para estudo da metodologia serverless, foram usados alguns serviços e ferramentas da Amazon Web Services. Foram usados 2 buckets S3, a ferramenta AWS Lambda e o serviço Rekognition. A arquitetura da aplicação pode ser visualizada na Figura 11. Conforme já descrito em capítulos anteriores, um bucket S3 possui a capacidade de ser utilizado para a hospedagem de sites estáticos. Um dos buckets da aplicação é utilizado justamente desta forma: nomeado como repo-site, ele é responsável por hospedar a página estática que exibe ao usuário os resultados da análise feita (fotos identificadas e percentual de similaridade).

Figura 12 – Listagem dos buckets criados na console da AWS



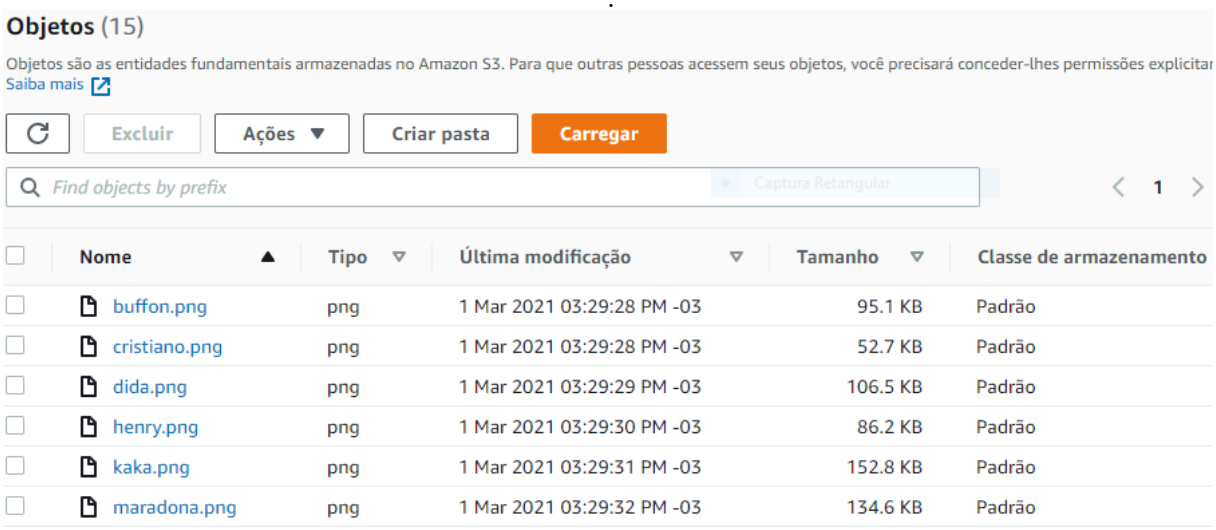
Fonte: Autor

Já o outro bucket S3, nomeado como repo-faces, possui duas funções na aplicação: inicialmente, foi povoado com as imagens utilizadas para a indexação prévia e criação da base de reconhecimento. Uma vez pronta a aplicação, esse mesmo bucket é responsável por receber os uploads com as imagens a serem analisadas pela função principal. A listagem dos buckets utilizados listados na console do Amazon S3 pode ser vista na Figura 12. A Figura 13 mostra as imagens carregadas no bucket repo-faces.

A ferramenta principal utilizada na aplicação foi o AWS Lambda. Conforme já dito anteriormente, o AWS Lambda é o serviço da Amazon que permite executar códigos sem provisionar servidores, e dispensando a necessidade de gerenciamento de recursos computacionais. O uso dessa ferramenta foi o responsável por garantir o funcionamento da aplicação conforme as definições básicas da metodologia serverless. Nesse serviço foi disponibilizada uma função chamada analiseFacial, desenvolvida em Python 3.7, que é responsável por controlar as ações principais da aplicação.

As aplicações sem servidor são orientadas a eventos. Ou seja, sua execução depende da ocorrência de determinados eventos pré-estabelecidos. Na nossa aplicação de estudo, essa característica se faz presente. O evento que ativa a execução da função analiseFacial é um upload no bucket de nome repo-faces. Dessa forma, sempre que uma nova imagem for inserida no bucket, esse código hospedado no Lambda será executado, havendo nesse momento apenas a concessão de recursos computacionais suficientes a esta execução.

Figura 13 – Imagens hospedadas no bucket repo-faces.



Fonte: Autor

Uma vez ocorrendo a execução da função hospedada no AWS Lambda, serão utilizados serviços de reconhecimento facial fornecidos pelo Amazon Rekognition, que fará a análise e devolverá os resultados à função, que será responsável também por disponibilizar estes resultados, de modo a serem utilizados pelo bucket que hospeda o site estático, e assim possam ser visualizados pelo usuário. Na próxima seção será feita uma breve descrição sobre o Rekognition e seus principais recursos. Vale ressaltar que alguns procedimentos, como uploads, foram feitos utilizando-se o serviço AWS CLI, que permite a manipulação via linha de comando das ferramentas da AWS. Na Figura 14 por exemplo, temos a visualização via AWS CLI da função lambda criada.

Figura 14 – Listagem via AWS CLI da função Lambda criada.

```
marco@marco:~$ aws lambda list-functions|grep -e FunctionName -e Runtime -e Mem
orySize -e Timeout
            "FunctionName": "analiseFacial",
            "Runtime": "python3.7",
            "Timeout": 3,
            "MemorySize": 128,
```

Fonte: Autor

4.2 Amazon Rekognition

Conforme explicado no tópico anterior, a aplicação em estudo utiliza, além das tecnologias S3 e Lambda, a tecnologia AWS Rekognition. Exemplo de SaaS (Software as a Service) o Rekognition é uma plataforma de visão computacional lançada em 2016. Fazendo

uso de Machine Learning, esse serviço oferece funcionalidades que permitem ao usuário a identificação de textos, pessoas, imagens, entre outros, fornecendo análise e pesquisa facial com alto grau de confiabilidade. Os recursos de visão computacional oferecidos pela plataforma podem ser divididos em 2 subgrupos: algoritmos que podem ser treinados pelo próprio usuário sobre um conjunto de dados definido previamente e personalizado, e algoritmos treinados previamente pela própria Amazon. Como exemplos destes últimos, temos recursos como reconhecimento de faces em celebridades, e a detecção/classificação de textos em imagens.

O AWS Rekognition tem vários recursos disponíveis ao usuário. Um deles é a identificação de rótulos, ou seja, análise de imagens visando identificar situações específicas, tais como objetos e cenas (como um estádio de futebol ou um shopping). Também possui um poderoso serviço de detecção de texto, podendo inclusive detectar palavras distorcidas ou inclinadas. Permite também promover moderação de conteúdo, sendo capaz de identificar conteúdos que podem ser considerados perigosos ou inadequados. Permite ainda a detecção, análise, pesquisa e confirmação de faces, podendo ainda mediante análise obter informações como sexo e faixa etária.

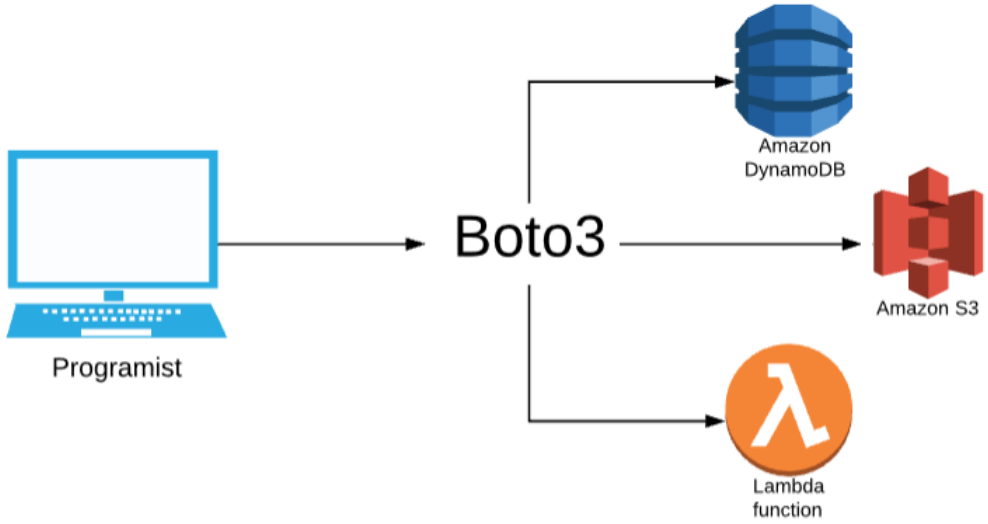
Uma das vantagens do uso da tecnologia AWS Rekognition é sua fácil integração com aplicações em geral, sejam mobile, desktop, ou qualquer outro tipo. Podendo ser operada inclusive por usuários sem conhecimento de Machine Learning e Deep Learning, é uma solução pronta para qualquer demanda que necessite de reconhecimento de imagens com alta precisão. Também oferece análise de imagens de forma altamente escalável, ou seja, independentemente da quantidade de imagens a serem reconhecidas, o serviço continua operando com alto desempenho e retornando tempos de respostas uniformes. Além disso, o serviço é totalmente integrável com todos os outros recursos da AWS, como Lambda e S3, tal qual foi usado na aplicação objeto de estudo do trabalho. Pode ser citado ainda como benefício o baixo custo a ser pago pela utilização do serviço, haja visto que os desenvolvedores serão cobrados apenas pelos recursos efetivamente consumidos (como quantidade de imagens analisadas).

Em relação á implementação, a aplicação em estudo faz uso de um SDK chamado boto3. Essa biblioteca, desenvolvida para ser usada em aplicações desenvolvidas na linguagem Python, permite a integração de aplicações com vários serviços da AWS, como Amazon EC2, CloudWatch e DynamoDB, conforme a Figura 15 demonstra. Na aplicação de análise facial, essa ferramenta é utilizada. Através do boto3 é que conseguimos, por exemplo, listar e ter acesso a todas as imagens do bucket repo-faces, para posterior indexação. Em relação aos buckets S3, o boto3 permite vários outros tipos de manipulação via Python, tais como upload de arquivos, download e configuração de permissões de acesso.

As funcionalidades do serviço AWS Rekognition utilizadas também foram acessadas

utilizando-se o SDK boto3. Na indexação de fotos, por exemplo, o boto3 nos oferece um método chamado index_faces, que é responsável por detectar rostos, e os adicionar a uma coleção. A coleção, neste caso, nada mais é do que um contêiner criado do lado do servidor, utilizado pelo Amazon Rekognition para armazenar informações referentes a faces detectadas. Diferente do que se possa pensar, o AWS Rekognition não salva os rostos reais detectados. O funcionamento consiste em um algoritmo de detecção, que após identificar os rostos, consegue extrair as características principais de cada um. Uma vez categorizadas as características determinantes na identificação de cada rosto, elas são salvas em um vetor de características, armazenadas num banco de dados, estando assim prontas para comparação com futuros casos de teste.

Figura 15 – SDK boto3 e alguns dos serviços acessíveis por ele.



.

Fonte: chaosgears (2019)

Além do index_faces, também são usados outros métodos, tais como list_faces (lista as faces em uma coleção), delete_faces (exclui uma ou mais imagens específicas de uma coleção) e o search_faces, que para um determinado rosto de entrada, compara o mesmo com todos os rostos indexados previamente em uma coleção e informa os resultados. Vale registrar que há um manual de utilização do SDK boto3 disponibilizado pela Amazon. Ele possui orientações de uso de dezenas de métodos relativos à análise e manipulação de imagens, possuindo explicações, detalhamento dos parâmetros de entrada e saída de cada função, e disponibiliza ainda um exemplo da sintaxe de implementação de cada função na linguagem Python, facilitando bastante o uso da tecnologia por desenvolvedores interessados, conforme demonstra a Figura 16.

4.3 Funcionamento da Aplicação em Estudo

Conforme já dito anteriormente, a aplicação utilizada como estudo de caso para análise da metodologia serverless se baseia numa função hospedada na ferramenta AWS

Figura 16 – Código Python referente à função index_faces, no manual do SDK boto3.

Request Syntax

```
response = client.index_faces(
    CollectionId='string',
    Image={
        'Bytes': b'bytes',
        'S3Object': {
            'Bucket': 'string',
            'Name': 'string',
            'Version': 'string'
        }
    },
    ExternalImageId='string',
    DetectionAttributes=[
        'DEFAULT'|'ALL',
    ],
    MaxFaces=123,
    QualityFilter='NONE'|'AUTO'|'LOW'|'MEDIUM'|'HIGH'
)
```

Fonte: Autor

Lambda, que utiliza os serviços de reconhecimento facial do serviço AWS Rekognition sobre fotos categorizadas previamente, sendo o evento de ativação da função o upload de uma foto num bucket S3. Para o funcionamento da aplicação, é preciso que haja previamente a seleção de fotos que irão compor essa base de comparação, e que seja promovida a indexação dessas fotos numa coleção, que é a estrutura usada pelo AWS Rekognition para armazenar as informações referentes a essa análise.

Para fins de teste, foram selecionadas 15 fotos de jogadores famosos de futebol. Inicialmente, foi feito o upload das imagens através do AWS CLI num bucket S3 de nome repo-faces, conforme demonstra a Figura 17.

Figura 17 – Upload das imagens selecionadas no bucket repo-faces.

```
marco@marco:~/Imagens$ aws s3 sync . s3://repo-faces
upload: ./messi.png to s3://repo-faces/messi.png
upload: ./buffon.png to s3://repo-faces/buffon.png
upload: ./ronaldo.png to s3://repo-faces/ronaldo.png
upload: ./kaka.png to s3://repo-faces/kaka.png
upload: ./cristiano.png to s3://repo-faces/cristiano.png
upload: ./zidane.png to s3://repo-faces/zidane.png
upload: ./neymar.png to s3://repo-faces/neymar.png
upload: ./dida.png to s3://repo-faces/dida.png
```

Fonte: Autor

Após o upload das fotos no bucket, o próximo passo foi a criação da coleção responsável por receber as informações referentes à indexação e à análise das fotos escolhidas.

A criação da coleção também foi feita utilizando-se a tecnologia AWS CLI, vide Figura 18.

Figura 18 – Criação da coleção via AWS CLI.

```
marco@marco:~$ aws rekognition create-collection --collection-id "faces_analise
"
{
    "StatusCode": 200,
    "CollectionArn": "aws:rekognition:us-east-1:621806537314:collection/faces_a
nalise",
    "FaceModelVersion": "5.0"
}
```

Fonte: Autor

O próximo passo seria efetuar de fato a indexação das fotos a serem analisadas na coleção recém-criada. Para isso, foi feito uso do SDK boto3, já comentado anteriormente. Através dele foi possível acessar via código as imagens disponibilizadas no bucket S3 repo-faces, conforme a Figura 19, e posteriormente indexá-las à coleção utilizando-se o método index_faces. As imagens utilizadas na indexação são mostradas na Figura 21.

Figura 19 – Código usado para acessar as imagens do bucket S3.

```
6   s3 = boto3.resource('s3')
7   def lista_imagens():
8       listagembucket=[]
9       bucket = s3.Bucket('repo-faces')
10      for aux in bucket.objects.all():
11          listagembucket.append(aux.key)
12
13      return listagembucket
```

Fonte: Autor

Após feitos os procedimentos descritos anteriormente, as imagens já estão indexadas na coleção, como mostra a listagem via AWS CLI na Figura 20. A aplicação já está apta a receber uma foto de teste e fazer a identificação de faces nela utilizando os serviços do AWS Rekognition e a devolver os resultados desta análise, tendo como base as faces categorizadas anteriormente. Nesse momento que entra em ação a função no AWS Lambda, que vai controlar todas estas ações. Essa função chama-se analiseFacial, e é implementada em Python 3.7. No processo de criação de uma função Lambda, o usuário é responsável por definir a memória alocada para ser consumida pela função em execução, e o tempo máximo de execução, conforme mostra a Figura 22.

Após a criação da função analiseFacial, foi configurado via console da AWS o evento de execução dela. Esse evento foi definido como sendo um upload de imagem no bucket S3

Figura 20 – Imagens já indexadas à coleção.

```
marco@marco:~$ aws rekognition list-faces --collection-id faces_analise |grep E
xternalImageId
            "ExternalImageId": "cristiano",
            "ExternalImageId": "zico",
            "ExternalImageId": "buffon",
            "ExternalImageId": "henry",
            "ExternalImageId": "zidane",
            "ExternalImageId": "ronaldo",
            "ExternalImageId": "romario",
            "ExternalImageId": "rooney",
            "ExternalImageId": "maradona",
            "ExternalImageId": "kaka",
            "ExternalImageId": "dida",
```

Fonte: Autor

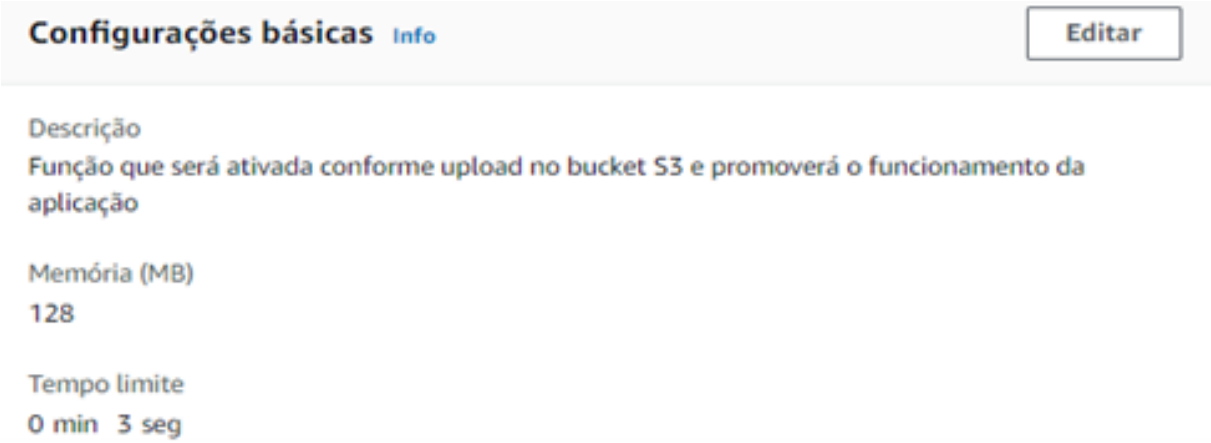
Figura 21 – Imagens carregadas no bucket S3 e utilizadas na indexação.

Fonte: Autor

repo-faces, conforme demonstra a Figura 23. Ou seja, sempre que uma nova imagem for carregada no bucket, a função analiseFacial será executada e fará a análise da imagem em questão. Vale ressaltar que é preciso definir uma política de recursos específica que permita ao AWS Lambda conseguir ter acesso ao serviço S3.

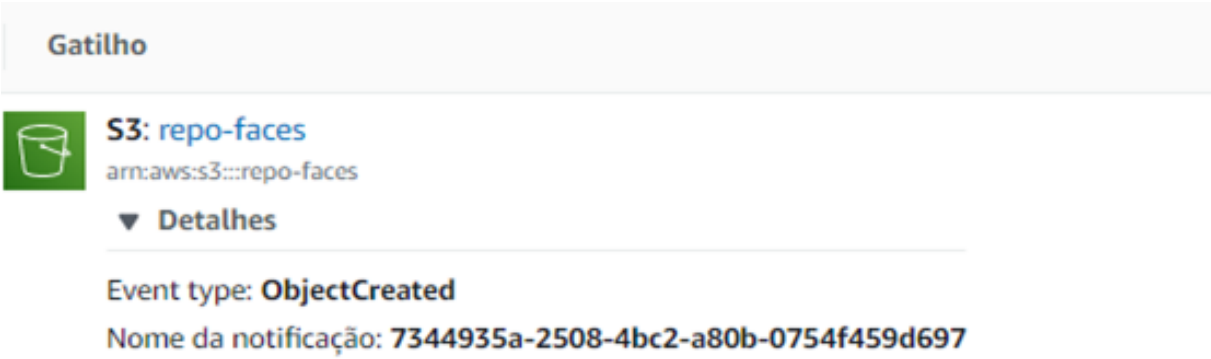
Uma vez criada e configurada a função, o upload de fotos no bucket S3 repo-faces é feito de forma a testar a aplicação. Esse carregamento de fotos para teste foi feito via linha

Figura 22 – Configurações básicas da função Lambda.



Fonte: Autor

Figura 23 – Evento de ativação da função Lambda configurado.



Fonte: Autor

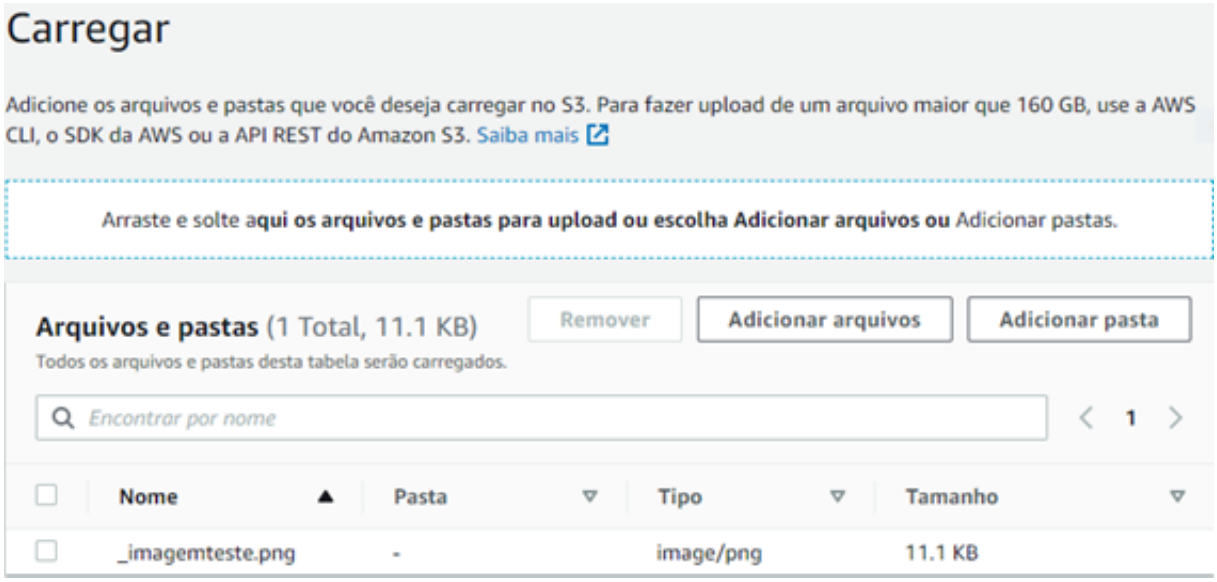
de comando, conforme mostrado na Figura 24. Vale ressaltar que esse upload também poderia ter sido feito diretamente na console do bucket S3 para fins de teste, conforme mostrado na Figura 25.

Figura 24 – Upload de uma foto de teste no bucket S3 via AWS CLI.

```
marco@marco:~/Imagens$ aws s3 cp _imagemteste.png s3://repo-faces
upload: ./_imagemteste.png to s3://repo-faces/_imagemteste.png
marco@marco:~/Imagens$
```

Fonte: Autor

Figura 25 – Upload de foto para fins de teste via console da AWS.

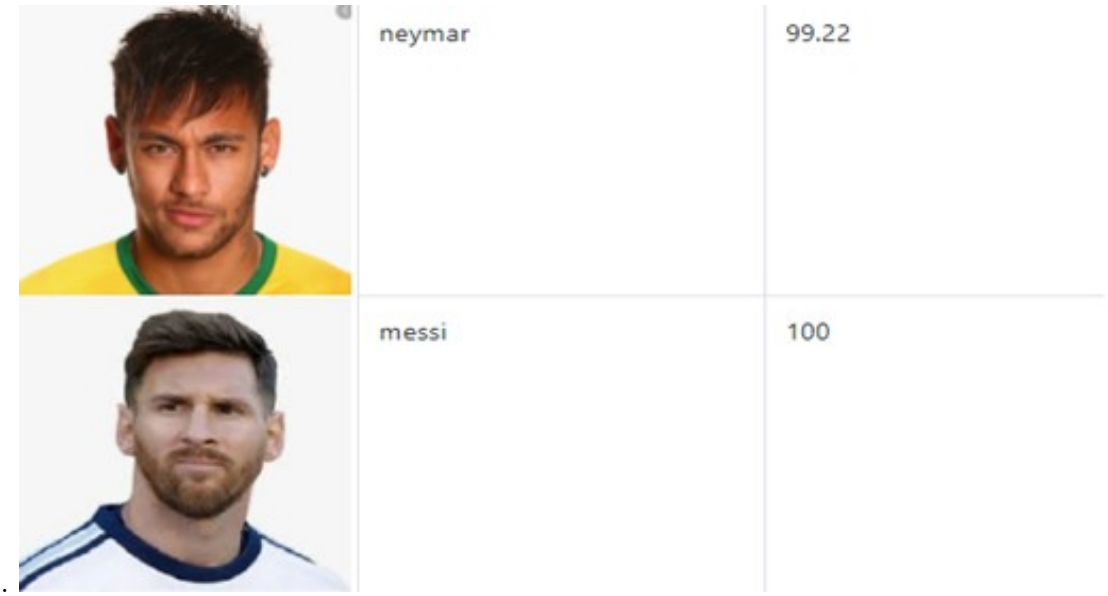


Fonte: Autor

5 RESULTADOS

Para efetuar os testes da aplicação, foram feitos uploads de fotos no bucket S3 repo-faces via linha de comando, utilizando-se AWS CLI. Após a execução de alguns desses uploads, foi atualizada a página estática hospedada no bucket repo-site, e verificou-se que o resultado mostrado já estava atualizado de acordo com a última imagem carregada no bucket, conforme mostrado nas Figuras 26 e 27. Ou seja, ficou demonstrada de fato a funcionalidade da função lambda, e sua execução controlada pelo evento de upload de uma imagem no bucket repo-faces. Na página é mostrada a foto categorizada previamente na coleção do AWS Rekognition, o nome do jogador e o percentual de similaridade em relação à foto de teste. A Figura 28 mostra as imagens de teste usadas na obtenção dos 2 resultados comentados anteriormente.

Figura 26 – Exemplo de resultado da análise.

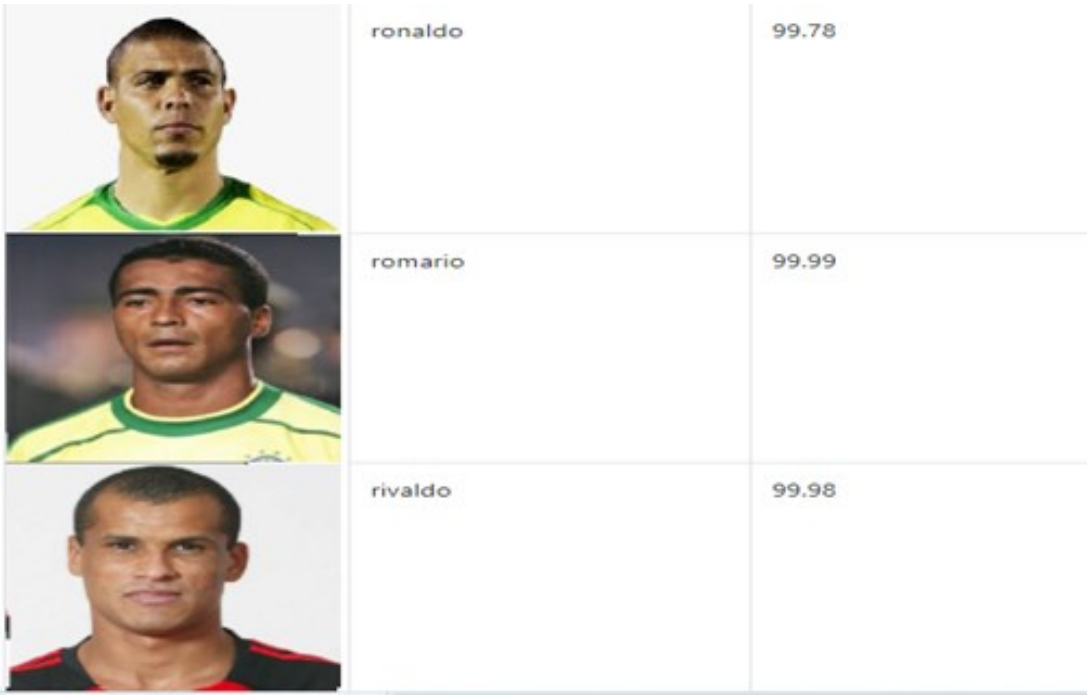


Fonte: Autor

O serviço AWS Lambda é fortemente integrado ao Amazon CloudWatch, que é o serviço de monitoramento de serviços em nuvem fornecido pela Amazon. Para cada execução da função Lambda, é gerado um log no CloudWatch, que uma vez detalhado fornece informações como resultado, duração da execução e memória consumida, conforme demonstram as Figuras 29 e 30 As informações de log também podem ser visualizadas na parte de monitoramento, dentro da console do próprio AWS Lambda, vide Figura 31.

Na aba de monitoramentos na console do AWS Lambda, também é possível visualizar vários gráficos com informações sobre as execuções da função, como número de invocações da função, e duração, para um dado intervalo de tempo, como demonstrado na

Figura 27 – Exemplo de resultado da análise.



Fonte: Autor

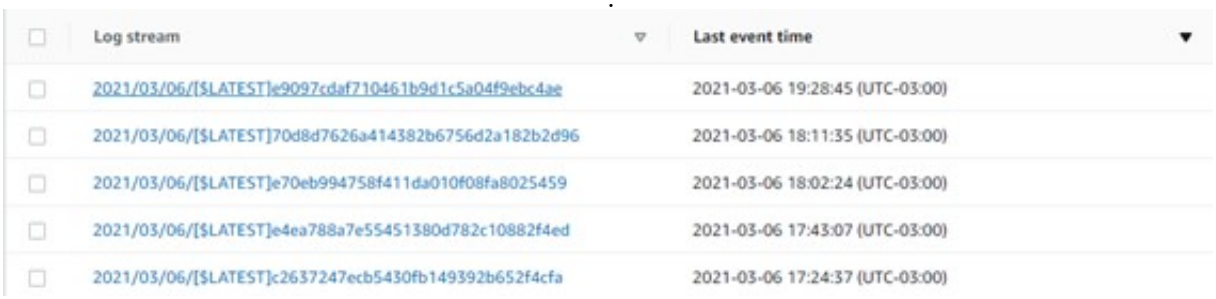
Figura 28 – Fotos utilizadas como teste para obtenção dos resultados anteriores.

Fonte: Autor

Figura 32. Também é possível adicionar esses gráficos a um painel do CloudWatch conforme a necessidade de análise do usuário. Essas informações podem ainda ser visualizadas nas métricas do CloudWatch, onde podemos visualizar valores mínimo, máximo e médias, à exemplo da Figura 33. As informações encontradas nas métricas também podem ser exportadas para visualização em um painel CloudWatch, conforme mostrado na Figura 34.

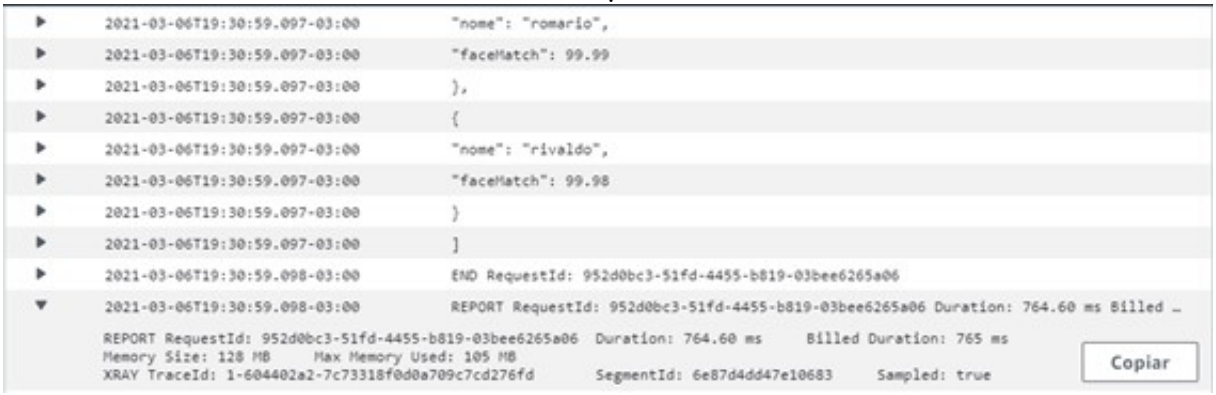
A AWS oferece outras ferramentas que nos permitem fazer análises construtivas sobre a execução de funções Lambda. Uma delas é a ferramenta Lambda Insights, que promove o levantamento e agregação de logs de performance e métricas coletados em

Figura 29 – Listagem de Logs de execução do Lambda no CloudWatch.



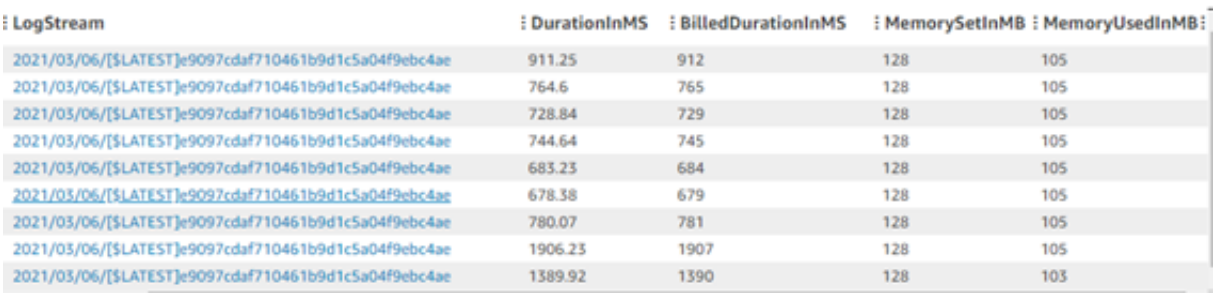
Fonte: Autor

Figura 30 – Detalhamento de um dos logs de execução da função Lambda.



Fonte: Autor

Figura 31 – Logs de execução visualizados dentro da console do AWS, informando duração das execuções e memória utilizada em cada uma delas.



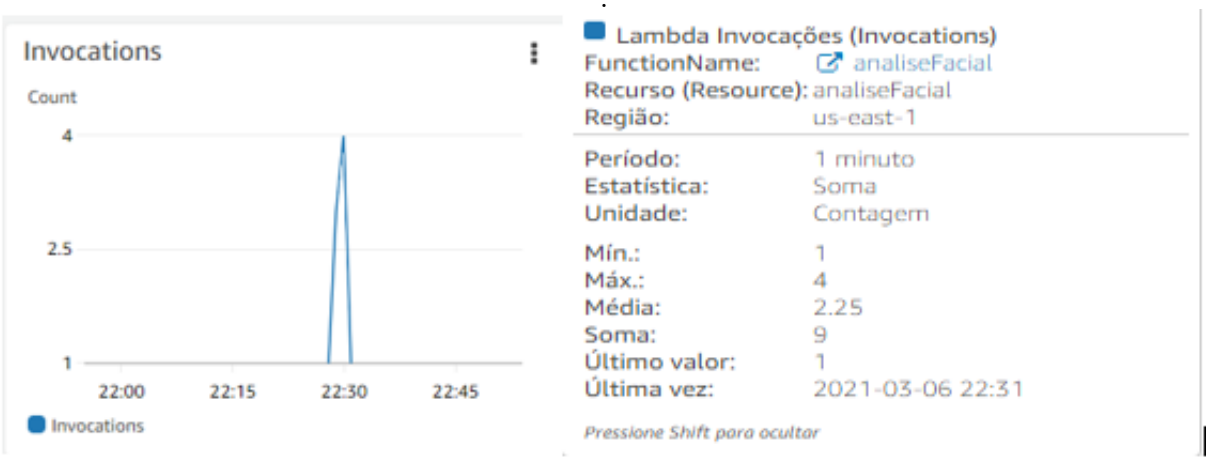
LogStream	DurationInMS	BilledDurationInMS	MemorySetInMB	MemoryUsedInMB
2021/03/06/[$LATEST]e9097cdaf710461b9d1c5a04f9ebc4ae	911.25	912	128	105
2021/03/06/[$LATEST]e9097cdaf710461b9d1c5a04f9ebc4ae	764.6	765	128	105
2021/03/06/[$LATEST]e9097cdaf710461b9d1c5a04f9ebc4ae	728.84	729	128	105
2021/03/06/[$LATEST]e9097cdaf710461b9d1c5a04f9ebc4ae	744.64	745	128	105
2021/03/06/[$LATEST]e9097cdaf710461b9d1c5a04f9ebc4ae	683.23	684	128	105
2021/03/06/[$LATEST]e9097cdaf710461b9d1c5a04f9ebc4ae	678.38	679	128	105
2021/03/06/[$LATEST]e9097cdaf710461b9d1c5a04f9ebc4ae	780.07	781	128	105
2021/03/06/[$LATEST]e9097cdaf710461b9d1c5a04f9ebc4ae	1906.23	1907	128	105
2021/03/06/[$LATEST]e9097cdaf710461b9d1c5a04f9ebc4ae	1389.92	1390	128	103

Fonte: Autor

tempo de execução. Através dele podemos ter acesso a métricas que detalham a execução e permitem o monitoramento e solução de problemas. Entre essas informações temos uso de disco, rede, tempo de CPU e percentual de memória utilizada. Um exemplo do uso dessa ferramenta é mostrado na Figura 35.

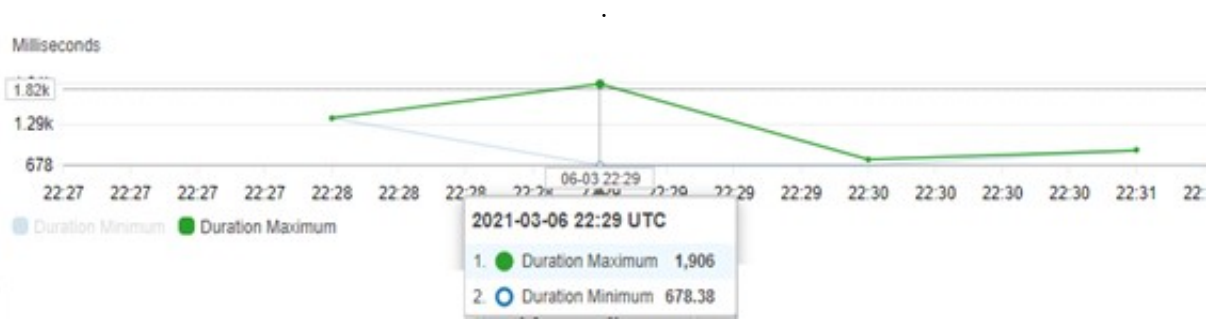
O AWS Lambda pode ainda ter seus resultados monitorados por outra ferramenta

Figura 32 – Gráfico referente ao número de invocações da função.



Fonte: Autor

Figura 33 – Informações de execução da função visualizadas nas métricas do CloudWatch, informando a duração máxima e mínima.

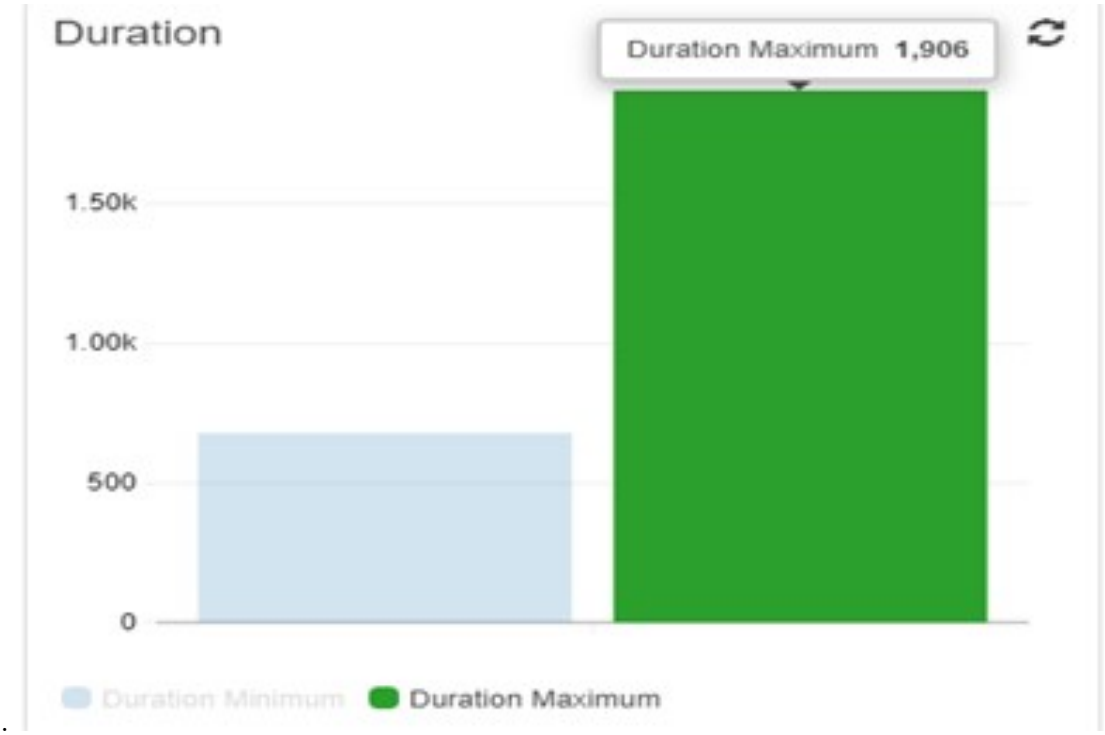


Fonte: Autor

chamada AWS X-Ray, que é capaz de visualizar os componentes do aplicativo, identificar gargalos de desempenho e solucionar problemas em geral. A ativação dessa ferramenta é feita pela própria console do AWS Lambda. Uma vez ativada, o serviço X-Ray receberá continuamente dados de rastreamento vindos do Lambda, e os transformará em dados prontos para análise do usuário. Os rastreamentos referentes a uma chamada também podem ser detalhados, nos oferendo informações como tempo de resposta da chamada e tempo de invocação. As figuras 36, 37 e 38 ilustram o uso da ferramenta na aplicação em estudo.

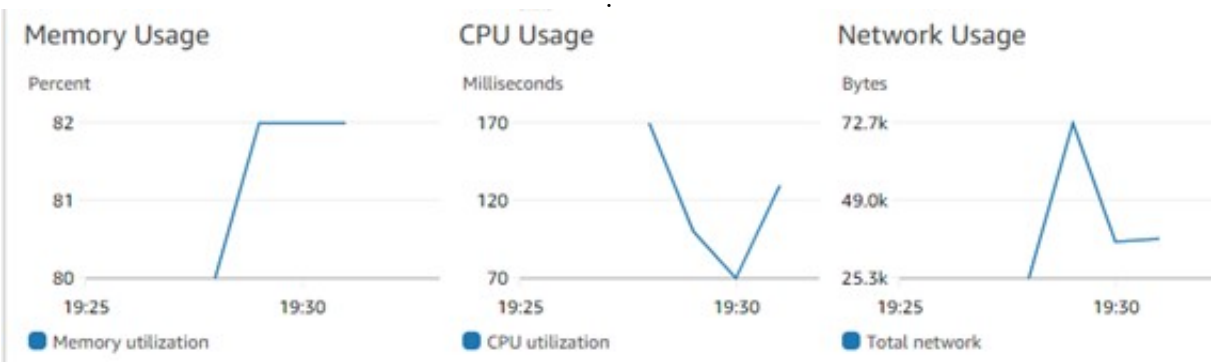
Dependendo do tipo de aplicação, uma solução serverless implementada com o uso do AWS Lambda pode ser mais vantajosa financeiramente ao usuário em comparação à uma solução que disponha do uso de instancias virtuais EC2. No Lambda, o pagamento a ser feito pelo usuário é calculado tendo como base apenas os recursos computacionais utilizados na execução das funções, não havendo, por exemplo, cobrança sobre tempo

Figura 34 – Informações sobre tempo de execução visualizadas num painel do CloudWatch.



Fonte: Autor

Figura 35 – Informações visualizadas no console da ferramenta Lambda Insights.

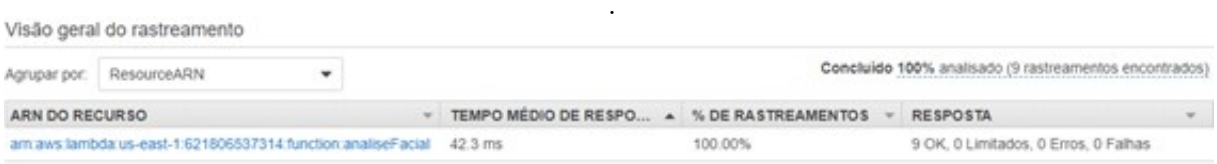


Fonte: Autor

ocioso, que pode acontecer em aplicações hospedadas no EC2 em momentos de total inatividade.

Nos testes da aplicação em estudo, foram feitos 9 uploads de imagem no bucket S3 repo-faces, eventos estes que fizeram a função analiseFacial ser executada 9 vezes. Nessas 9 execuções houve o consumo de 943 MB de memória, e o tempo de execução total foi de 8.587,16 milissegundos, conforme somatório feito sobre os valores registrados nos logs de execução na Figura 31. Tendo como base esses dados, podemos considerar os valores de 104,77 MB de memória consumida e de 954,13 milissegundos como sendo os valores

Figura 36 – Resumo dos rastreamentos de execução da função analiseFacial.



Fonte: Autor

Figura 37 – Diagrama mostrando tempo de execução e de resposta referente a uma invocação.

Fonte: Autor

Figura 38 – Chamada de função Lambda detalhada entre tempo de execução e de invocação.

Fonte: Autor

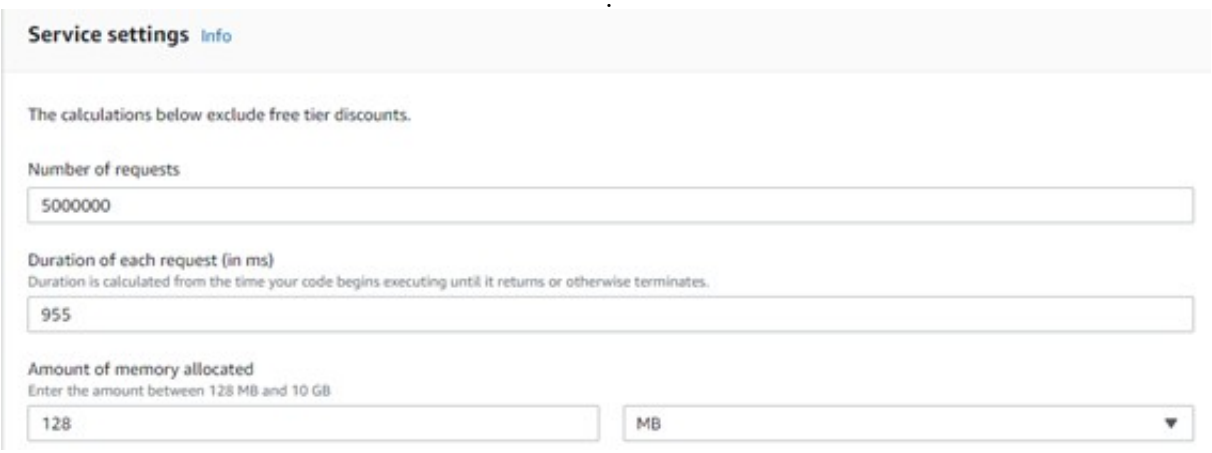
médios de execução da função em estudo. Tendo como base esses valores, foi feita uma análise de custo comparativa entre o uso do AWS Lambda e do Amazon EC2.

Para calcular esses custos, foi usada a ferramenta AWS Princing Calculator. Ela permite ao usuário criar uma estimativa financeira referente aos seus recursos e plataformas utilizadas na AWS. Através dela é permitido ao usuário modelar as soluções tecnológicas que deseja desenvolver antes de iniciar o processo de desenvolvimento, explorar os preços que podem vir a ser cobrados, e assim encontrar os tipos de serviços mais adequados à suas necessidades e à sua disponibilidade financeira. Ela permite fazer estimativas financeiras sobre todos os principais produtos da AWS, como EC2, Lambda, S3, Amazon RDS, entre outros.

Para a análise de custos referentes à ferramenta AWS Lambda, foi considerada uma média de tempo de execução de 955 milissegundos, conforme calculado nos testes

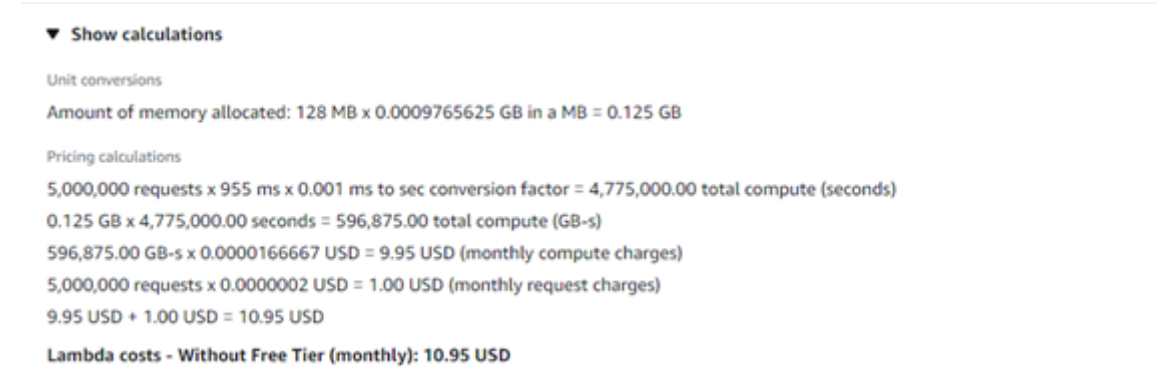
da aplicação em estudo. Quanto à memória alocada, o mínimo disponível é 128 MB. Considerando que nossa média de memória consumida foi inferior a 105MB, este valor foi escolhido na simulação. Quanto à quantidade de simulações, foi considerada uma situação em que a função será executada 5 milhões de vezes. A Figura 39 mostra os parâmetros utilizados na simulação em questão.

Figura 39 – Configuração dos parâmetros para análise de custo do AWS Lambda.



Fonte: Autor

Figura 40 – Resultado da análise de custos para os parâmetros definidos anteriormente.



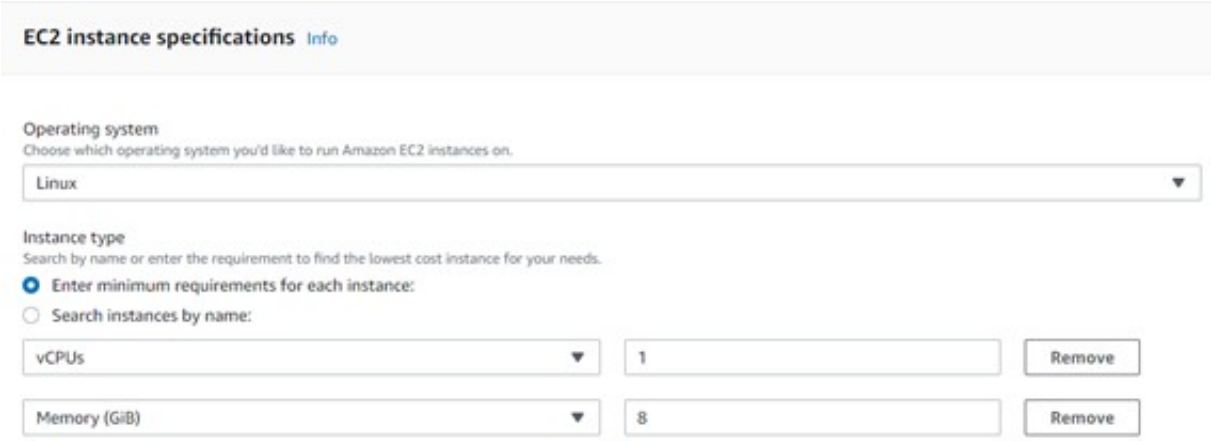
Fonte: Autor

Ou seja, para os parâmetros definidos, verificamos que teríamos um custo referente a 10.95 dólares, conforme demonstrado na Figura 40. Agora é preciso simular os custos referentes à uma instância EC2 e fazer o comparativo entre os valores. Para a simulação referente à criação de uma instância virtual EC2, foi considerada uma máquina relativamente simples, com sistema operacional Linux, 8 GB de memória e apenas um núcleo de processamento, conforme demonstrado na Figura 41.

Se ao invés de utilizarmos uma solução serverless, instanciássemos uma máquina EC2 para rodar uma aplicação do tipo, em uma máquina relativamente simples, seria preciso

Figura 41 – Configurações para simulação de preço do Amazon EC2.

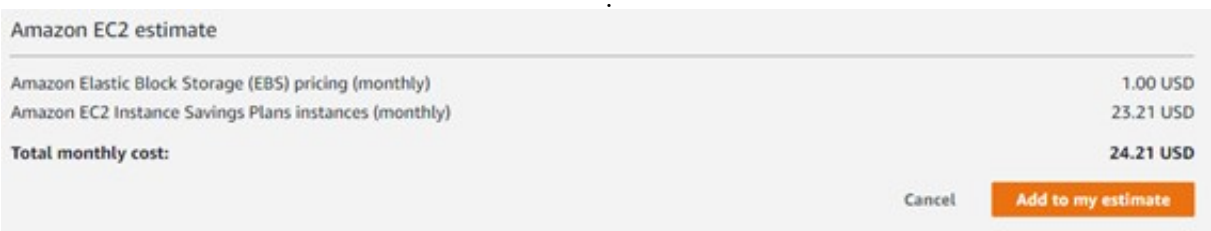
.



Fonte: Autor

Figura 42 – Resultado da simulação de preços referente ao Amazon EC2.

.



Fonte: Autor

arcar com um custo referente à 24.21 dólares mensais, de acordo com o resultado mostrado na Figura 42. Em comparação com a simulação feita referente ao uso do AWS Lambda, conclui-se que uma solução feita com base em instâncias EC2 se mostra financeiramente menos vantajosa.

6 CONCLUSÃO E TRABALHOS FUTUROS

De um modo geral, podemos afirmar que apesar do trabalho feito demandar um nível considerável de pesquisa, tendo em consideração o fato de abordar um tema contemporâneo ainda em evolução, e o estudo de caso demandar um certo grau de cuidado em relação ao uso das ferramentas de computação em nuvem disponibilizadas pela AWS, que devem sempre ser utilizadas com as devidas precauções para não ocasionar em gastos não previstos, aborda um tema central de grande relevância à área, haja visto a tendência cada vez maior no uso de soluções baseadas na abordagem serverless. Vale ressaltar a excelente funcionalidade do AWS Lambda em fornecer uma visão detalhada sobre os recursos computacionais utilizados nas execuções e de sua integração com serviços de monitoramento como o Amazon CloudWatch, fornecendo assim informações cruciais para a montagem da análise de custos feita e pro desenvolvimento do trabalho de um modo geral.

A análise do funcionamento da aplicação utilizada como estudo de caso nos mostra a eficiência da metodologia serverless e da ferramenta AWS Lambda no tocante à implementação sem uso de servidores ou instâncias virtuais, sua execução mediante ativação de eventos pré estabelecidos, e à concessão de recursos computacionais provisionada pelo provedor de serviços em nuvem(no caso a Amazon) de acordo com a demanda apresentada pela aplicação em questão.

Os resultados obtidos através da comparação de custos foram bastante satisfatórios, provando que, para o estudo de caso feito no trabalho, o uso da metologia serverless e das ferramentas abordadas, tal qual o AWS Lambda, se mostra economicamente mais vantajoso em comparação ao uso de outras ferramentas, como o Amazon EC2 ou qualquer outra tecnologia que provenha servidores físicos ou virtuais.

De um modo geral podemos afirmar que os resultados almejados neste trabalho foram alcançados. É preciso ressaltar que a vantagem da metologia serverless em relação à metodologias tradicionais, ou em detrimento ao uso de instâncias virtuais não é uma regra, podendo existir situações em que o uso de outras soluções se mostre mais vantajoso. Cabe ao desenvolvedor sempre estar analisando as características e particularidades de seus projetos para melhor definir a melhor abordagem de implementação. Como trabalhos futuros, podemos sugerir um estudo da situação comentada anteriormente, ou seja, situações em que o uso da metodologia serverless se mostre desvantajoso em relação a outras abordagens. Também pode-se sugerir estudos de caso utilizando outras ferramentas voltada para a computação sem servidor disponíveis no mercado, tais como o Azure Functions e o Cloud Functions.

Referências

AMAZON. *AmazonAWS Lambda Execute código sem se preocupar com servidores ou clusters. Pague somente pelo que usar*. 2014. Disponível em: <https://aws.amazon.com/pt/lambda/>. Acesso em: 16/02/2021. Citado na página 27.

AMAZON. *AmazonAmazon CloudWatch Capacidade de observação dos seus recursos da AWS e aplicativos na AWS e no local*. 2019. Disponível em: <https://aws.amazon.com/pt/cloudwatch/>. Acesso em: 12/02/2021. Citado na página 28.

AMAZON. *AmazonAmazon EC2 Capacidade de computação segura e redimensionável para oferecer suporte a praticamente qualquer carga de trabalho*. 2020. Disponível em: <https://aws.amazon.com/pt/ec2>. Acesso em: 14/02/2021. Citado na página 31.

AMAZON. *AmazonSem servidor na AWS Crie e execute aplicações sem se preocupar com servidores*. 2020. Disponível em: <https://aws.amazon.com/pt/serverless/>. Acesso em: 04/02/2021. Citado 3 vezes nas páginas 21, 28 e 30.

ANALYTICSHUT. *List S3 buckets using Python, AWS CLI*. 2021. Disponível em: <https://analyticshut.com/list-s3-buckets-using-python-aws-cli/>. Acesso em: 06/02/2021. Citado na página 26.

BACELAR P. POPIOLEK, V. H. J. T. N. D. O. M. E. *O Modelo de Computação em Nuvem e sua Aplicabilidade*. 2012. Citado na página 18.

CHAOSGEARS. *New challenges with Boto3*. 2019. Disponível em: <https://chaosgears.com/new-challenges-with-boto3/>. Acesso em: 20/02/2021. Citado na página 38.

DAROLT FELIPE RODRIGO DE SOUZA, G. P. K. D. D. *Explorando a elasticidade de nuvens IaaS para reconfigurar dinamicamente aplicações n-camadas*. 2016. Citado na página 18.

EDUCOUTINHO. *Como criar instância EC2 no Amazon AWS - Linux*. 2017. Disponível em: <https://educoutinho.com.br/amazon-aws/como-criar-instancia-ec2-linux/>. Acesso em: 09/02/2021. Citado 2 vezes nas páginas 32 e 33.

EDUREKA. *Amazon CloudWatch – A Monitoring Tool By Amazon*. 2020. Disponível em: <https://www.edureka.co/blog/amazon-cloudwatch-monitoring-tool/>. Acesso em: 08/02/2021. Citado na página 31.

ESTABIL. *AWS na prática | Zonas de disponibilidade*. 2018. Disponível em: <https://blog.estabil.is/aws-na-pratica-zonas-de-disponibilidade/>. Acesso em: 03/02/2021. Citado na página 24.

INTEGRATE, B. *MODELOS DE COMPUTAÇÃO EM NUVEM*. 2016. Disponível em: <http://blog.integrate.com.br/2016/08/modelos-de-computacao-em-nuvem.html>. Acesso em: 01/02/2021. Citado na página 19.

LIMA, G. D. O. *ARQPED: ARQUITETURA DE COMPUTAÇÃO EM NUVEM COM DEPENDABILIDADE*. 2014. Citado na página 17.

MICROSOFT. *Microsoft O que é computação em nuvem?* 2015. Disponível em: <https://azure.microsoft.com/pt-br/overview/what-is-cloud-computing/>. Acesso em: 30/01/2021. Citado na página 14.

MICROSOFT. *Microsoft O que é a computação sem servidor?* 2019. Disponível em: <https://azure.microsoft.com/pt-br/overview/serverless-computing/>. Acesso em: 05/02/2021. Citado na página 22.

OKITA TIAGO A. COIMBRA, C. B. R. E. B. N. T. *Desenvolvimento de ferramenta para otimização de custo na AWS*. 2018. Citado 3 vezes nas páginas 23, 25 e 27.

RIGHI, R. da R. *Elasticidade em cloud computing:conceito, estado da arte e novos desafios*. 2013. Citado 2 vezes nas páginas 15 e 16.

SILVA, J. C. Q. da. *USO DE NUVENS HÍBRIDAS NAS EMPRESAS BRASILEIRAS*. 2017. Citado 2 vezes nas páginas 19 e 20.

SUITE, I. *Computação em Nuvem: Inovações em TI para impulsionar os negócios*. 2019. Disponível em: <https://www.iperiusbackup.net/pt-br/computacao-em-nuvem-inovacoes-em-ti-para-impulsionar-os-negocios/>. Acesso em: 03/02/2021. Citado na página 21.

TELECO. *Cloud Computing: Computação em Nuvem*. 2018. Disponível em: <https://www.teleco.com.br/tutoriais/tutorialcloudcaracten/pagina_2.asp>. Acesso em: 31/01/2021. Citado na página 14.

WIKIPEDIA. *Computação em nuvem*. 2021. Disponível em: <https://pt.wikipedia.org/wiki/Computaç~ao_em_nuvem>. Acesso em: 01/02/2021. Citado na página 17.