INSTITUTO FEDERAL DE SANTA CATARINA

ANDERSON GASPAR DE MEDEIROS

Análise de possibilidades de implantação de uma aplicação baseada em microsserviços em multinuvem para redução de custos de operação

São José - SC

Dezembro/2023

Anderson Gaspar de Medeiros
Análise de possibilidades de implantação de uma aplicação baseada em microsserviços em multinuvem para redução de custos de operação/ Anderson Gaspar de Medeiros. – São José - SC, Dezembro/2023-
40 p. : il. (algumas color.) ; 30 cm.

Orientador: Prof. Ederson Torresini, Msc.

Monografia (Graduação) – Instituto Federal de Santa Catarina – IFSC
Campus São José
Engenharia de Telecomunicações, Dezembro/2023.

1. Computação em nuvem. 2. Multinuvem. 2. Custos de operações. I. Ederson Torresini. II. Instituto Federal de Santa Catarina. III. Campus São José. IV. Análise de possibilidades de implantação de uma aplicação *Web* em multinuvem para redução de custos de operação.

ANDERSON GASPAR DE MEDEIROS

ANÁLISE DE POSSIBILIDADES DE IMPLANTAÇÃO DE UMA APLICAÇÃO BASEADA EM MICROSSERVIÇOS EM MULTINUVEM PARA REDUÇÃO DE CUSTOS DE OPERAÇÃO

Este trabalho foi julgado adequado para obtenção do título de Engenheiro de Telecomunicações, pelo Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina, e aprovado na sua forma final pela comissão avaliadora abaixo indicada.

São José - SC, 14 de Dezembro de 2023:

Prof. Ederson Torresini, Msc.
Orientador
Instituto Federal de Santa Catarina

Prof. Jorge Henrique Busatto Casagrande, Dr.
Instituto Federal de Santa Catarina

Prof. Cleber Jorge Amaral, Dr.
Instituto Federal de Santa Catarina

AGRADECIMENTOS

Gostaria de expressar meus sinceros agradecimentos a todas as pessoas que desempenharam um papel crucial na realização deste trabalho e que foram fundamentais na minha jornada acadêmica e profissional.

Primeiramente, à minha esposa, Fábia, minha companheira de vida, cujo apoio inabalável e crença constante neste sonho foram essenciais para alcançar este objetivo.

À minha família, meus pais, Antonio e Irene, e minha irmã, Iara, quero expressar minha profunda gratidão. Seu amor incondicional, investimento em minha educação e apoio inabalável ao longo de toda a minha vida foram fundamentais para meu crescimento pessoal e acadêmico.

Ao meu orientador, Ederson Torresini, sou grato pelo apoio constante que sempre disponibilizou. Sua orientação entusiástica foi fundamental para o desenvolvimento deste trabalho e para o meu crescimento como estudante e pesquisador.

Ao IFSC (Instituto Federal de Santa Catarina), minha instituição de ensino, agradeço por proporcionar um ambiente de aprendizado de alta qualidade e pelas oportunidades de capacitação que me foram oferecidas ao longo de minha jornada acadêmica. Isso foi fundamental para minha formação e para meu ingresso bem-sucedido no mercado de trabalho.

Por fim, gostaria de expressar minha gratidão a todas as pessoas que, de alguma forma, contribuíram para a realização deste trabalho, mesmo que não tenham sido mencionadas aqui. Suas contribuições, diretas ou indiretas, foram verdadeiramente valiosas.

A todos vocês, meu sincero obrigado por fazerem parte desta jornada e por tornarem este trabalho uma realidade. Suas contribuições e apoio significaram muito para mim.

"Aprender é a única coisa que a mente nunca se cansa,
nunca tem medo e nunca se arrepende. "
(Leonardo da Vinci)

RESUMO

A computação em nuvem vem proporcionando que aplicações sejam disponibilizadas na internet com alta escalabilidade e estabilidade para qualquer tipo de demanda. Hoje há vários provedores que disponibilizam o serviço de computação em nuvem e é importante que empresas saibam onde melhor alocar os recursos computacionais gastando menos. Este trabalho realiza um estudo de caso focado na implementação de uma aplicação baseada em microsserviços em multinuvem explorando distintos provedores e abordagens para uma análise da variação dos custos operacionais. A pesquisa ofereceu perspectivas sobre as possibilidades de implementação de recursos computacionais visando a redução de despesas operacionais.

Palavras-chave: Computação em nuvem. Multinuvem. Custos de operações.

ABSTRACT

Cloud computing has enabled applications to be deployed on the internet with high scalability and stability to meet any kind of demand. Today, there are several providers offering cloud computing services, and it is crucial for businesses to know where to best allocate computational resources while minimizing costs. This study focuses on a case analysis centered around the implementation of a microservices-based application in a multicloud environment, exploring different providers and approaches to analyze the variation in operational costs. The research provides insights into the implementation possibilities of computational resources with the aim of reducing operational expenses.

Keywords: Cloud computing. Multicloud. Operations costs.

LISTA DE ILUSTRAÇÕES

Figura 1 – Modelos de infraestrutura 17
Figura 2 – Cenário 1 31
Figura 3 – Cenário 2 31
Figura 4 – Cenário 3 32
Figura 5 – Cenário 4 33
Figura 6 – Cenário 5 34
Figura 7 – Cenário 6 35
Figura 8 – Cenário 7 36

LISTA DE TABELAS

Tabela 1 – Preços das instâncias 29

LISTA DE ABREVIATURAS E SIGLAS

CNCF *Cloud Native Computing Foundation* 23

DNS *Domain Name System* 24

EKS *Amazon Elastic Kubernetes Service* 30

FaaS *Function-as-a-Service* 18

IaaS *Infrastructure-as-a-Service* 13

NAS *Network Attached Storage* 21

PaaS *Platform-as-a-Service* 17

SaaS *Software-as-a-Service* 17

SAN *Storage Area Network* 21

SO *System Operation* 22

VM *Virtual Machine* 20

AWS *Amazon Web Services* 18

Azure *Azure Cloud Platform* 22

GCP *Google Cloud Platform* 22

RDS *Amazon - Relational Database Service* 33

S3 *Amazon - Simple Storage Service* 32

ECS *Amazon - Elastic Container Service* 30

SQL *Structured Query Language* 28

EBS *Amazon - Elastic Block Store* 32

VPN *Virtual private network* 32

HTTP *Hypertext Transfer Protocol* 25

EC2 *Amazon - Elastic Compute Cloud(EC2)* 32

USD *USD - United States dollar* 30

REST *REST - Representational State Transfer* 27

SOAP *SOAP - Simple Object Access Protocol* 27

CPU *CPU - Central processing Unit* 17

RAM *RAM - Random Access Memory* 24

CIPS *CIPS - Gartner Cloud Infrastructure and Platform Services* 27

SUMÁRIO

1 INTRODUÇÃO . . . 13
1.1 Objetivo Geral . . . 14
1.2 Metodologia . . . 14
1.3 Organização do texto . . . 15

2 FUNDAMENTAÇÃO TEÓRICA . . . 16
2.1 História da Nuvem . . . 16
2.2 O que é computação em nuvem . . . 16
2.3 Modelos de infraestrutura . . . 17
2.4 Modelos de implantação . . . 18
2.4.1 Nuvem pública . . . 18
2.4.2 Nuvem privada . . . 19
2.4.3 Nuvem híbrida - multinuvem . . . 19
2.5 Tipos de virtualização . . . 20
2.5.1 Virtualização de rede . . . 20
2.5.2 Virtualização de armazenamento . . . 21
2.5.2.1 NAS . . . 21
2.5.2.2 SAN . . . 21
2.5.3 Virtualização de processamento . . . 21
2.5.3.1 *Full virtualization* . . . 21
2.5.3.2 *Paravirtualization* . . . 22
2.6 Contêineres . . . 22
2.6.1 Docker . . . 23
2.6.2 Orquestração de Contêineres com Kubernetes . . . 23
2.6.3 Escalabilidade Horizontal e Vertical . . . 24
2.7 Descoberta de Serviço em uma Arquitetura Cliente-Servidor . . . 24
2.7.1 Registro centralizado . . . 24
2.7.2 Uso de *proxy* ou *gateway* . . . 25
2.7.3 Descoberta de serviço baseada em *DNS* . . . 25
2.8 *Serverless* . . . 25

3 DESENVOLVIMENTO . . . 27
3.1 Provedores de Serviços em Nuvem . . . 27
3.1.1 Amazon Web Services (AWS) . . . 27
3.1.2 Microsoft Azure . . . 27

3.1.3 Google Cloud Platform (GCP) 28
3.2 Comparação de Preços sob Demanda entre as Principais Plataformas de Nuvem: *AWS*, *Azure* e *GCP* **28**
3.3 Aplicação baseada em Arquitetura de Microsserviços **29**
3.4 Cenários de Investigação **30**
3.4.1 Cenários *IaaS* 30
3.4.2 Cenários *PaaS* 32
3.4.3 Multinuvem 34
3.4.3.1 *Service Mesh* 35

4 CONCLUSÕES **37**
4.1 Trabalhos futuros **38**

REFERÊNCIAS **39**

1 INTRODUÇÃO

As inovações tecnológicas da informação e comunicação nas últimas décadas vêm afetando como as empresas de rápido crescimento estruturam seus modelos de negócios Hamel e Prahalad (1996) argumentam que competências centrais são desenvolvidas por meio de um processo de melhoramentos contínuos ao longo do tempo e que, para se ter sucesso no mercado global, é mais importante construir competências centrais do que integrar verticalmente.

A computação vem se tornando uma aliada na construção e crescimento das empresas juntamente com a evolução da internet. Foi a partir de 1994 que a internet ampliou suas funções para ser mais que uma rede de circulação de informações para entrar na comercialização de produtos e serviços. Com isso abriu-se a possibilidade para investir em novas tecnologias que pudessem atender melhor as empresas e/ou consumidores, umas das tecnologia é a computação em nuvem.

A ideia de computação em nuvem começou na década de 1960 com alguns expoentes dessa época, alguns deles eram cienticistas do Instituto de Tecnologia de Massachusetts Mccarthy (1960), nove anos antes da internet ser criada. Nos anos 1990, já com a *World Wide Web*, houve avanços nos sistemas de telecomunicações que abriram as portas para o futuro da computação em nuvem. Em 1997 foi usado pela primeira vez o termo *Cloud Computing* por pesquisadores da universidade do Texas em Austin Chellappa, Barua e Whinston (1997). Dois anos depois a empresa Salesforce foi a primeira a disponibilizar aplicações empresariais pela Web.

No início desta década, empresas como Google e Amazon criaram imensos parques computacionais baseados no conceito de nuvem para operar seus negócios de forma independente umas das outras. É a origem do que hoje chamamos de *Infrastructure-as-a-Service* (IaaS). Uma vez tendo desenvolvido estas imensas infraestruturas, descobriram que poderiam permitir o acesso remoto ao processamento, possibilitando que outras empresas eliminassem o investimento inicial em *hardware*. Assim passaram a ofertar seus serviços de computação em nuvem em escala global, materializando os trabalhos dos acadêmicos dos anos 1960 em diante.

Apesar dos serviços ofertados por meio da computação em nuvem, a questão financeira para as empresas é um fator determinante para implementação dos negócios. Os serviços em nuvem tratam de serviços por demanda e podem impactar positivamente na redução de custos. É o primeiro passo para implementar novas tecnologias, pois automatiza o uso de recursos computacionais, com economia e gestão inteligente de processo. É uma alternativa sustentável para suprir as demandas de tecnologia da informação da

empresa. As pequenas empresas, em particular, estão recorrendo a esses serviços de armazenamentos de documentos e de rodar um processador de textos para evitar a complexidade de operação e manutenção de manter seus próprios centros de processamento de dados. Outra forma é de computação terceirizada, que envolve o fornecimento de poder de processamento e capacidade de armazenamento.

Reduzir os custos e aumentar a produtividade é o desejo de toda empresa atualmente. No mercado globalizado e competitivo, qualquer diferencial no negócio é uma vantagem perante os concorrentes. Aumentar a produtividade através da computação na nuvem pode ser um diferencial no mercado. Computação em nuvem aumenta a velocidade de troca de informações entre setores de uma empresa por meio de ferramentas colaborativas, possibilitando trabalhar de forma integrada sem precedentes.

Segundo Richards (2017) microserviços são uma estrutura arquitetônica que fragmenta uma aplicação em unidades independentes, cada qual encarregada de uma função específica. Hoje as aplicações baseada em microsserviços são geralmente executadas no tipo cliente-servidor, pois tem-se um cenário acessível, possibilita troca de mensagens pela rede. É uma vantagem adicional, fazer o uso de uma arquitetura de microsserviços, para esse tipo de arquitetura facilitando a manutenção. A execução ocorre parte no servidor e parte no cliente, cliente entende-se como usuário, na manutenção é possível substituir, reparar, atualizar ou mesmo realocar de um servidor sem afetar o processo, servidor esse que pode estar em outra nuvem pública, isso pode implicar custo de operação menor.

1.1 Objetivo Geral

Este trabalho tem como objetivo investigar as possibilidades de implantação de uma aplicação baseada em microsserviços em ambiente multinuvem visando redução de custos de operação.

1.2 Metodologia

Este trabalho contém 4 etapas para alcançar o objetivo. Na primeira etapa tem-se o estudo sobre os diferentes tipos de computação em nuvem, sua origem, modelos de serviço e implantação, modelos de virtualizações. Na segunda etapa serão avaliado os possíveis cenários de implantação de uma aplicação com uma arquitetura de microsserviços, onde esses microsserviços poderão ser implantados em serviços específicos ou serviços genéricos em grandes provedores de nuvem pública, sempre visando no menor custo de operação. Na terceira etapa será feito uma análise de custos de operação dos diferentes cenários. Na quarta etapa será feito a conclusão dos estudos dos cenários e seus custos de operação.

1.3 Organização do texto

O texto está organizado da seguinte forma: No Capítulo 1 é apresentada a introdução do trabalho com a descrição da proposta, motivação e objetivo. No Capítulo 2 é apresentado uma revisão teórica sobre os conceitos gerais sobre computação em nuvem, modelos de implantação e diversos tipos de virtualização de rede, processamento e armazenamento. No Capítulo 3, é dedicado à discussão dos provedores de serviços em nuvem. Realizamos uma análise comparativa de preços entre as plataformas líderes, fundamentando as escolhas das instâncias para hospedagem da aplicação web cliente-servidor. Adicionalmente, exploramos a implementação dessa aplicação com uma arquitetura de microsserviços, destacando suas vantagens e adaptações específicas para o escopo do projeto. No Capítulo 4, concluímos este trabalho sintetizando as descobertas, enfatizando as contribuições realizadas e delineando possíveis direções para pesquisas futuras. Essa seção proporciona uma visão consolidada do projeto, destacando a relevância dos elementos discutidos ao longo do texto.

2 FUNDAMENTAÇÃO TEÓRICA

A Fundamentação teórica abordará conceitos sobre computação em nuvem, tipos de virtualização, modelos de infraestrutura, ideia de infraestrutura como código e contêineres.

2.1 História da Nuvem

No início dos tempos da computação, basicamente o que se tinha eram grandes computadores, chamados de *mainframes* onde ficavam o processamento das tarefas e tinham os terminais, chamados "terminais sem processamento local" que era essencialmente uma tela e um teclado, sem processamento local para o usuário. Então com a redução dos custos dos computadores, surgiram os primeiros minicomputadores por volta de 1970, popularmente conhecidos como computadores pessoais, os *PCs*, esses tinham processamento local. Com a disseminação dos *PCs*, começaram as primeiras conexões entre *PCs*, então surgiu o conceito de Redes, onde máquinas poderiam conversar entre elas em uma ainda pequena rede de intranet, por exemplo em universidades e empresas.

Esse conceito de redes foi expandido para a internet, agora máquinas fornecem e recebem serviços de outras redes, por exemplo o modelo cliente-servidor. Nesse conceito também evoluiu para a Computação em *Grid*, onde percebeu-se que tinha muito processamento ocioso, então porque não contratar esse processamento, porque não disponibilizar para terceiros esse processamento para processar os dados.

A nuvem é fundamentalmente uma computação em *Grid* com uma grande quantidade de gerenciamento extra, onde é possível alocar e desalocar recursos, um ambiente mais dinâmico, pelo qual você paga pelo que você usa, na forma de modelo sob demanda.

2.2 O que é computação em nuvem

Nuvem pode ser definida como um grande reservatório de recursos computacionais (processamento, rede, dados, software, hardware etc.) que é um sistema construído de forma escalável e acessível via rede e sob demanda e a ideia da computação em nuvem é usar esses recursos para construir sistemas. Trata-se de um paradigma em que o sistema é projetado para usar recursos remotos com controle de acesso.

A computação em nuvem, de acordo com NIST (2011) é um modelo para habilitar o acesso por rede ubíquo, conveniente e sob demanda a um conjunto compartilhado de recursos de computação (como redes, servidores, armazenamento, aplicações e servi-

ços) que possam ser rapidamente provisionados e liberados com o mínimo de esforço de gerenciamento ou interação com o provedor de serviços.

As características essências da computação em nuvem são: agilidade e elasticidade, confiabilidade e tolerância a falhas, serviço mensurável, transparência do reservatório de recursos. Com a agilidade e elasticidade é possível alocar e desalocar recursos sob demanda de forma automática ou manual. A tolerância a falhas refere-se a redundâncias de dados, fica mais difícil perder dados e também é orientada a serviço, ou seja, falha de um serviço não afeta outros serviços.

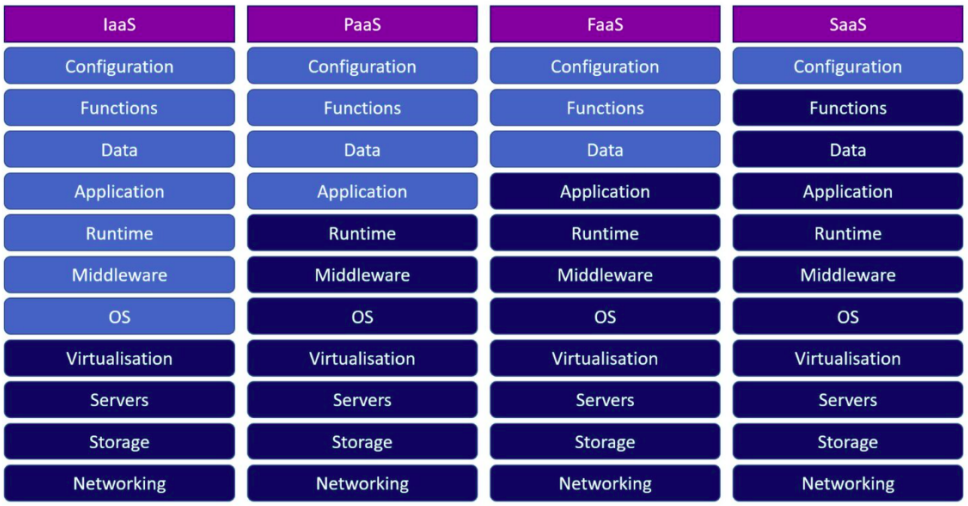
Outro aspecto importante é que você paga pelo que você usa, então o serviço deve ser mensurável, precisa prover métricas de uso, como por exemplo, banda, *CPU - Central processing Unit* (CPU), disco, transações e também de forma de assinatura mensal ou anual. No contexto da transparência do reservatório de recursos, refere-se ao fato de que a alocação de recursos físicos e virtuais ocorre de forma opaca ao usuário. Em outras palavras, não é essencial saber exatamente em qual equipamento os recursos estão operando, desde que os requisitos operacionais sejam devidamente atendidos.

A seguir será tratado sobre modelos de infraestrutura, modelos de implantação, tipos de virtualização, infraestrutura como código, contêineres, *serverless* e observabilidade e monitoramento.

2.3 Modelos de infraestrutura

Os quatro grandes modelos de infraestrutura são: Infraestrutura como Serviço ou IaaS, Plataforma como Serviço ou *Platform-as-a-Service* (PaaS) e Software como Serviço ou *Software-as-a-Service* (SaaS). Na Figura 1 estão ilustrados os modelos.

Figura 1 – Modelos de infraestrutura



Fonte: Analytics (2021)

IaaS refere-se ao fornecimento de infraestrutura computacional como um serviço. É fornecido apenas uma infraestrutura básica(máquina virtual, com interface de rede, armazenamento). O usuário final precisa configurar e gerenciar a plataforma e o ambiente, implantar aplicativos. Na Figura 1, é possível observar que as camadas de virtualização, servidores, armazenamento e rede são de responsabilidade do provedor.

PaaS é fornecido uma plataforma que permite ao usuário final desenvolver, executar e gerenciar aplicativos sem a complexidade de construir e manter a infraestrutura. Nesse caso toda a parte operacional e software da infraestrutura correspondente já está preparada para o usuário. Comumente esse modelo é mais usado por desenvolvedores. Na Figura 1, é provável notar que a responsabilidade do usuário é das camadas de dados e aplicação e as outras camadas são de responsabilidade do provedor.

SaaS também chamado de "*software* sob demanda", normalmente acessados por usuários finais por meio de navegador web ou pela *interface* de *moblie apps*. No SaaS, tudo pode ser gerenciados pelos fornecedores como, aplicativos, dados, *middleware*, sistemas operacionais, virtualização, armazenamento e rede. É possível verificar na Figura 1 que o usuário só precisa executar a aplicação no provedor da nuvem, todas as outras camadas são de responsabilidade do provedor de serviço.

Function-as-a-Service (FaaS) permite o gerenciamento das funções de aplicativos, eliminando a necessidade dos usuário gerenciarem a infraestrutura, o sistema operacional ou os aplicativos. Isso não apenas alivia o usuário(desenvolvedor) dos encargos de gerenciamento (como PaaS), mas também permite muita escalabilidade. A cobrança é feita por chamada de função ou tempo de execução, com uma granularidade fina, o que poderia ser muito benéfico para novos serviços.

2.4 Modelos de implantação

Os modelos de implantação de uma infraestrutura em nuvem podem ser classificadas em três grupos: nuvem pública, privada e híbrida.

2.4.1 Nuvem pública

Na nuvem pública, segundo o padrão estabelecido pelo NIST (2011), a infraestrutura da organização é mantida por uma empresa externa, um provedor de serviços, como a *Amazon Web Services* (AWS). Os serviços são acessados pela Internet. O provedor de serviços é responsável por gerenciar e manter a infraestrutura física. Esse modelo de nuvem é altamente versátil, atraindo uma ampla gama de clientes, desde *startups* até grandes empresas. Ele se baseia no princípio de serviços cobrados com base na demanda e no uso, o que proporciona eficiência financeira ao evitar recursos ociosos. Além disso, a nuvem pública oferece excelente capacidade de escalabilidade para atender às necessida-

des variadas de seus usuários. Essa flexibilidade e adaptabilidade tornam a nuvem pública uma escolha popular para uma variedade de perfis de clientes.

2.4.2 Nuvem privada

A nuvem privada, conforme definido pelo NIST (2011), é um modelo de implantação no qual a infraestrutura de nuvem é provisionada exclusivamente para uso por uma única organização, que pode incluir múltiplos consumidores, como diferentes unidades de negócios dentro da empresa. Essa infraestrutura pode ser de propriedade, gerenciada e operada pela própria organização, por terceiros ou por uma combinação de ambos.

A característica fundamental da nuvem privada é a exclusividade de uso, garantindo que todos os recursos, dados e configurações estejam sob controle direto da organização. Essa abordagem é amplamente adotada por instituições financeiras, como bancos, devido a regulamentações estritas e à necessidade de manter um alto nível de controle sobre a infraestrutura.

A nuvem privada oferece à organização um nível elevado de personalização e configuração sob medida para atender às suas necessidades específicas. Isso é essencial para garantir a conformidade com regulamentações rigorosas e a manutenção da segurança dos dados. A capacidade de gerenciar internamente a infraestrutura confere à organização maior autonomia e flexibilidade na administração de seus recursos de nuvem.

2.4.3 Nuvem híbrida - multinuvem

A nuvem multinuvem, de acordo com a definição do NIST (2011), é um modelo de implantação que permite combinar os benefícios dos modelos de nuvem pública e nuvem privada. Isso implica que parte do processamento ocorre na nuvem pública, enquanto outra parte ocorre na nuvem privada. Por exemplo, em instituições financeiras, a interação com o usuário, como páginas da web, pode ser hospedada na nuvem pública, enquanto operações transacionais críticas são executadas na nuvem privada.

Além disso, ao implementar uma arquitetura de microsserviços em uma abordagem de nuvem híbrida, é fundamental considerar outros elementos essenciais. A exportação de *endpoints*, por exemplo, permite que os microsserviços se comuniquem efetivamente entre ambientes de nuvem, tornando possível a distribuição de tarefas em diferentes plataformas. Os *gateways*, por sua vez, atuam como pontos de entrada para a rede de microsserviços, facilitando o roteamento e o controle de tráfego entre nuvens públicas e privadas. Além disso, a implementação de um *service mesh* melhora a comunicação entre os microsserviços da aplicação web, oferecendo recursos de balanceamento de carga, monitoramento e gerenciamento de tráfego.

Vale mencionar que a abordagem de multinuvem (ou *multicloud*) difere da nuvem híbrida, embora ambas envolvam o uso de várias nuvens. Na multinuvem, duas ou mais nuvens distintas (privada, pública ou híbrida) permanecem como entidades únicas, porém, são interligadas por tecnologias padronizadas ou proprietárias que possibilitam a portabilidade de dados e aplicativos, como o *cloud bursting* para equilíbrio de carga entre nuvens.

Essas distinções são cruciais para entender como diferentes arquiteturas em nuvem podem ser aplicadas com eficácia, de acordo com as necessidades e os requisitos específicos de cada organização.

2.5 Tipos de virtualização

A virtualização na computação é, em essência, abstrair a camada inferior e apresentar uma versão reduzida/modificada para a camada superior.

O principal objetivo da virtualização é gerenciar a carga de trabalho transformando a computação tradicional mais escalável, eficiente e econômica. A virtualização pode ser aplicada a uma ampla gama de aplicações, como virtualização de sistema operacional, virtualização em nível de hardware e virtualização de servidores.

Para este estudo será analisado virtualização de redes, armazenamento e processamento.

2.5.1 Virtualização de rede

A virtualização de rede destina-se a criar várias redes virtuais acima de uma rede física compartilhada, cada uma pode ser implantada e gerenciada de forma independente. A virtualização de rede refere-se à subdivisão de recursos de rede, hardware e software. Pode ser usada para consolidar muitas redes físicas, subdividir essas redes ou conectar máquinas virtuais *Virtual Machine* (VM).

Essa abordagem transforma as redes estáticas e inflexíveis em redes dinâmicas, ágeis e otimizadas. A arquitetura da rede física, incluindo *switches* e roteadores, é unificada e torna-se acessível para todos os usuários por meio de um sistema de gerenciamento centralizado. A virtualização de rede também permite a automação de muitas tarefas administrativas, reduzindo erros manuais e o tempo de aprovisionamento. Isso resulta em maior produtividade e eficiência na rede.

Um exemplo de virtualização de rede podemos citar a *VLAN* que é uma subseção de uma rede local (*LAN*), esse exemplo permiti a implantação de múltiplas redes virtuais sob uma extensa rede física.

A virtualização de redes pode ser do tipo externa ou interna. A virtualização de rede externa utiliza uma unidade virtual para combinar sistemas fisicamente ligados à mesma rede local (*LAN*) em redes locais virtuais (VLANs) separadas. A virtualização de rede interna atua dentro de um servidor utilizando contêineres de software para fornecer a funcionalidade de uma única rede física.

2.5.2 Virtualização de armazenamento

Uma camada de software é criada entre os discos físicos e os dispositivos que acessam esses discos, de modo a tornar o acesso mais flexível, personalizável e gerenciável. A seguir é apresentado dois tipos de virtualização de armazenamento *Network Attached Storage* (NAS) e *Storage Area Network* (SAN).

2.5.2.1 NAS

O NAS consiste em um computador com unidade principal e armazenamento que está conectado a uma rede. Esta unidade de armazenamento contém um ou mais discos rígidos com sistemas operacionais com menos recursos. A unidade principal é responsável pelo sistema de arquivos. NAS fornece compartilhamento de dados independentemente do sistema operacional. Para aumentar a disponibilidade, várias cópias de arquivos são armazenadas em localização múltipla. Também reduz a largura de banda necessária e hora de acessar. Os dados são acessados por metadados de arquivo no *NAS* Jaikar et al. (2016).

2.5.2.2 SAN

SAN fornece sistema de armazenamento de alta velocidade capaz de criar e utilizar uma rede exclusiva entre o *Storage* e os Servidores apenas para o armazenamento, com leituras e gravações em blocos. O compartilhamento de arquivos na *SAN* depende do sistema operacional. Na *SAN*, os dados são acessados por bloco que é replicado para vários locais para aumentar a disponibilidade. Jaikar et al. (2016)

2.5.3 Virtualização de processamento

Há dois tipos principais: *full virtualization* e *paravirtualization*.

2.5.3.1 *Full virtualization*

Na *full virtualization*, existe o *hypervisor* fornece ao sistema operacional visitante uma camada inferior e isolada do hardware subjacente. O *hypervisor* é responsável por carregar o convidado sistemas operacionais. E cada sistema operacional convidado será executado como se estivessem operando diretamente no hardware subjacente. Ou seja,

cada sistema operacional convidado terá todos os recursos do hardware subjacente. Os *hypervisors* que suportam a *full virtualization* têm um console de gerenciamento de máquina virtual do qual cada máquina virtual convidada pode ser facilmente gerenciada (BABU et al., 2014). O gasto computacional deste tipo de virtualização é alto e, em muitos casos, é impraticável manter esse nível de virtualização. Uma vantagem desse modelo de virtualização é a não necessidade de alterar o SO virtualizado.

2.5.3.2 *Paravirtualization*

A *paravirtualization*, é uma técnica na qual o sistema operacional está ciente de que está sendo operado diretamente em o *hypervisor* em vez da camada inferior do hardware . No *paravirtualization* um *hypervisor* é instalado no sistema operacional *host* que é executado na camada inferior do hardware. Este *hypervisor* atuará como uma camada de virtualização. Ou seja, o *hypervisor* atuará como o *host* no qual os sistemas operacionais convidados são carregados. Sistemas operacionais convidados farão as chamadas de sistema necessárias para hypervisores para a utilização de recursos de hardware Babu et al. (2014). Diferente da *full virtualization*, se faz necessário alterar o *System Operation* (SO) para que ocorra a comunicação com o *hypervisor*.

Esses são os principais tipos de virtualização, cada um com suas vantagens e desvantagens, e são amplamente utilizados em ambientes de data centers e nuvem para otimizar a utilização de recursos e melhorar a flexibilidade e eficiência operacional.

É importante observar que provedores de nuvens, como AWS, *Azure Cloud Platform* (Azure) e *Google Cloud Platform* (GCP), podem adotar modelos de precificação variados para serviços de virtualização. Eles podem cobrar pelos recursos alocados, pelo uso efetivo desses recursos ou por uma combinação de ambos. Os detalhes específicos das políticas de preços são determinados por cada provedor de nuvem e podem estar sujeitos a mudanças ao longo do tempo. Portanto, é fundamental que organizações e usuários estejam cientes das estruturas de preços dos provedores de nuvem ao implementar soluções de virtualização, a fim de gerenciar efetivamente os custos e otimizar os investimentos.

2.6 Contêineres

A virtualização baseada em contêineres, como exemplificado pelo *Docker*, representa um avanço significativo em relação à *full virtualization*, pois elimina a necessidade de recursos adicionais, como múltiplos sistemas operacionais, resultando em maior eficiência e desempenho (Cloud Google, 2022). Ao contrário da *full virtualization*, onde cada máquina virtual requer seu próprio sistema operacional e *hypervisor*, os contêineres compartilham um único *kernel* do sistema operacional, o que os torna extremamente leves e eficientes.

2.6.1 Docker

O *Docker*, desenvolvido como uma plataforma de código aberto, oferece uma gama de ferramentas, incluindo o *Docker CLI* e o *Docker Engine*, que gerenciam contêineres e seu isolamento de sistemas operacionais. O uso do *Docker* simplifica o empacotamento e a distribuição de aplicativos juntamente com suas dependências, aumentando a produtividade e reduzindo os custos operacionais (BERNSTEIN, 2014). Além disso, os contêineres *Docker* ajudam a mitigar divergências entre equipes de desenvolvimento e operações, permitindo que desenvolvedores se concentrem no desenvolvimento de aplicativos e as equipes de infraestrutura, frequentemente chamadas de *DevOps*, gerenciem a infraestrutura subjacente.

Embora o *Docker* seja uma escolha popular, é importante observar que existem alternativas viáveis, como o *containerd*, que também fornecem funcionalidades de gerenciamento de contêineres eficientes e confiáveis.

2.6.2 Orquestração de Contêineres com Kubernetes

De acordo com Kubernetes (2019), *Kubernetes*, também conhecido como K8s, é uma plataforma de orquestração de contêineres de código aberto que automatiza tarefas críticas, como implantação, dimensionamento e gerenciamento de contêineres. Ele oferece recursos avançados, incluindo balanceamento de carga, auto-escalonamento e recuperação automática de falhas. O projeto de código aberto é mantido e hospedado pela *Cloud Native Computing Foundation* (CNCF).

O *Kubernetes* opera através da criação de objetos *Kubernetes* por meio de uma *API*. Esses objetos são utilizados para definir as características de um aplicativo, como o tipo de aplicativo, as cargas de trabalho associadas, as imagens de contêiner, o número de réplicas, os recursos de armazenamento, e outras opções cruciais para a configuração e gestão do estado de um *cluster*. Uma vez que a configuração é estabelecida, o Painel de Controle do *Kubernetes* opera de forma contínua para garantir que o estado atual do *cluster* em execução esteja em conformidade com o estado configurado. Isso envolve a execução de ações automatizadas, como a inicialização e reinicialização de contêineres, bem como o dimensionamento do número de réplicas de uma aplicação específica (KUBERNETES, 2019).

Um *cluster* gerenciado pelo *Kubernetes* consiste em duas principais unidades operacionais: o mestre e os nós. Os componentes mestres disponibilizam serviços por meio de uma *API* e são responsáveis por executar tarefas relacionadas ao escalonamento, agendamento e atualizações de aplicações. A arquitetura do *Kubernetes* segue o modelo mestre-escravo, onde os mestres são responsáveis pelo gerenciamento, configuração e manutenção dos nós que compõem o *cluster*. Os nós de execução, por sua vez, são as unidades onde

as aplicações são implantadas, utilizando a virtualização de contêineres para garantir isolamento e eficiência (KUBERNETES, 2019).

2.6.3 Escalabilidade Horizontal e Vertical

Na gestão de recursos e desempenho de aplicativos em microsserviços, dois conceitos fundamentais são a escalabilidade horizontal e vertical.

- **Escalabilidade horizontal**: refere-se ao ato de adicionar mais instâncias idênticas de um componente, como um servidor ou contêiner, para lidar com um aumento na carga de trabalho. É uma estratégia eficaz para lidar com flutuações de tráfego e garantir que a aplicação permaneça responsiva. No contexto de contêineres e *Kubernetes*, isso pode ser alcançado facilmente adicionando mais contêineres em um *cluster* para distribuir a carga.
- **Escalabilidade vertical**: envolve aumentar os recursos de uma única instância, como adicionar mais CPU, *RAM - Random Access Memory* (RAM) ou outros recursos, para melhorar o desempenho. Isso é mais apropriado quando a carga de trabalho é previsível e as mudanças na demanda são relativamente suaves.

Ambos os tipos de escalabilidade desempenham papéis essenciais na otimização de desempenho e na gestão de custos em ambientes que fazem uso de microsserviços. Nessa arquitetura, a escalabilidade se torna ainda mais perceptível devido ao tamanho e funcionalidade de cada aplicação. A escolha entre escalabilidade horizontal e vertical depende das necessidades específicas do aplicativo e das condições de uso, sendo fundamental para garantir um desempenho ideal e manter os custos operacionais sob controle.

2.7 Descoberta de Serviço em uma Arquitetura Cliente-Servidor

A Descoberta de Serviço é uma prática essencial para garantir que os clientes encontrem e se comuniquem com os servidores apropriados em uma aplicação distribuída.

2.7.1 Registro centralizado

Registro centralizado envolve a criação de um servidor de registro central que mantém um registro atualizado de todos os servidores disponíveis na arquitetura. Os servidores registram suas informações, como endereços IP e portas, neste servidor central. Quando um cliente precisa localizar um serviço, ele consulta esse registro central para obter as informações necessárias. Essa abordagem é eficaz e amplamente utilizada, pois o *Domain Name System* (DNS) é uma parte fundamental da infraestrutura da Internet.

2.7.2 Uso de *proxy* ou *gateway*

A estratégia de Uso de *Proxy* ou *Gateway* envolve a introdução de um componente intermediário, como um *proxy* ou *gateway*, entre os clientes e os servidores. Esse intermediário é responsável por rotear as solicitações dos clientes para os servidores apropriados com base em vários critérios, como carga de trabalho, localização geográfica ou disponibilidade.

Essa abordagem oferece uma maior flexibilidade na decisão de roteamento, permitindo que critérios personalizados determinem para qual servidor uma solicitação deve ser direcionada. Além disso, o *proxy* ou *gateway* pode lidar com tarefas adicionais, como autenticação e criptografia, antes de encaminhar a solicitação para o servidor real.

2.7.3 Descoberta de serviço baseada em *DNS*

A Descoberta de Serviço Baseada em DNS faz uso do sistema de nomes de domínio (DNS) para direcionar as solicitações dos clientes para servidores específicos. Cada servidor é associado a um nome de domínio ou subdomínio. Quando um cliente precisa acessar um serviço, ele consulta o DNS para obter o endereço IP associado a esse nome de domínio. Isso direciona a solicitação do cliente para o servidor correto. Essa abordagem é eficaz e amplamente utilizada, pois o DNS é uma parte fundamental da infraestrutura da Internet.

2.8 *Serverless*

Serverless é um serviço em nuvem orientado a eventos onde é englobado o uso de serviços de nuvem sem a necessidade de provisionamento de recurso computacional prévio. Usuário paga apenas cada vez que seu código é executado (também conhecido como *FaaS*). O termo *serverless* é devido ao usuário precisar apenas escrever o código Mujezinović e Ljubović (2019). Onde o código é armazenado e como ele é executado é tratado pelo fornecedor. Além disso, o escalonamento sem servidor é feito automaticamente, embora outros serviços em nuvem possam fornecer escalabilidade ilimitada, o cliente possuirá certas ações para facilitar isso, como programar dentro de aplicativos ou entrar em contato com o fornecedor. Visto que o *Serverless* será dimensionado sem qualquer responsabilidade do cliente.

O *Google App Engine*, lançado em 2008, foi a primeira oferta de computação abstrata sem servidor. O *App Engine* incluiu funções *Hypertext Transfer Protocol* (HTTP) com um tempo limite de 60 segundos e um armazenamento de *blobs* e armazenamento de dados com seus próprios tempos limite.

O *AWS* Lambda, introduzido pela *Amazon* em 2014, popularizou o modelo abs-

trato de computação sem servidor. Ele é compatível com várias ferramentas adicionais sem servidor da *AWS*, como o modelo de aplicativo sem servidor da *AWS Amazon CloudWatch* e outras.

3 DESENVOLVIMENTO

No atual cenário de mercado, a concorrência entre fornecedores de serviços em nuvem é grande, com muitas opções disponíveis. No entanto, para os propósitos desta pesquisa, direcionamos nossa atenção primordialmente para apenas para alguns, conforme Saraswat e Tripathi (2020) e Kaushik et al. (2021). Vale ressaltar que essa seleção é consistente com a posição dessas plataformas no Quadrante Mágico da Gartner para *CIPS - Gartner Cloud Infrastructure and Platform Services* (CIPS) Amazon (2022b). A seguir, exploraremos cada uma dessas plataformas.

3.1 Provedores de Serviços em Nuvem

No atual mercado, há uma ampla concorrência entre diversos fornecedores de serviços em nuvem. No entanto, nosso foco será principalmente na AWS, Azure e GCP, pois esses provedores oferecem amplos serviços em nuvem e possuem uma presença significativa na indústria segundo as pesquisas realizadas pelos Saraswat e Tripathi (2020) e Kaushik et al. (2021). Vamos examinar cada uma dessas plataformas com mais detalhes.

3.1.1 Amazon Web Services (AWS)

A AWS, um dos pioneiros e provedores mais antigos de serviços em nuvem, lançou sua plataforma baseada em nuvem em 2006. Ao longo dos anos, a AWS consolidou sua posição como uma concorrente proeminente, capturando uma parcela significativa do mercado tanto em IaaS (47,8%) quanto em PaaS (34%) até o final de 2019. A AWS oferece uma ampla gama de serviços, incluindo armazenamento baseado em nuvem, computação e funcionalidades diversas, permitindo que os usuários configurem aplicativos e serviços sob demanda utilizando os protocolos *REST - Representational State Transfer* (REST) e *SOAP - Simple Object Access Protocol* (SOAP) sobre HTTP. Com sua presença global e uma extensa variedade de produtos, a AWS atende às necessidades de empresas em busca de serviços confiáveis e estáveis Saraswat e Tripathi (2020).

3.1.2 Microsoft Azure

Lançado em 2010, o Azure surgiu como uma alternativa na prestação de serviços em nuvem. No mercado de SaaS, o Azure lidera com uma participação de mercado de 17%, além de ocupar a segunda posição tanto em PaaS quanto em IaaS, com uma participação de mercado de 15,5% até o final de 2019. O Azure oferece aos clientes a flexibilidade de aproveitar os serviços em nuvem exclusivamente na nuvem ou em combinação com apli-

cativos, data centers ou infraestrutura existentes. As ofertas do Azure incluem um ambiente baseado em *Windows* para implantação de aplicativos, serviços de dados baseados no *Structured Query Language* (SQL) *server* e serviços *.NET* para infraestrutura distribuída. Com seu ecossistema voltado para aplicativos baseados na *Microsoft/Windows*, o Azure atende às necessidades de organizações que operam nesse ambiente Saraswat e Tripathi (2020).

3.1.3 Google Cloud Platform (GCP)

Com uma taxa de crescimento notável tanto em IaaS (60,2%) quanto em PaaS (57,9%) até o final de 2019, o GCP se destaca como uma das empresas que mais crescem no mercado de serviços em nuvem. Embora tenha ingressado no mercado em 2008, inicialmente oferecendo serviços como e-mail, calendários e documentação online, o GCP evoluiu para uma plataforma de hospedagem de aplicativos da web. Sua abordagem inclui o uso de um ambiente de tempo de execução pré-definido, especialmente voltado para o desenvolvimento e implantação de aplicativos baseados em nuvem, com suporte para linguagens de programação como *Java* e *Python*. O GCP é impulsionado por diversos fatores que contribuem para sua popularidade, como sua escalabilidade, infraestrutura baseada em usuários e opções de pagamento flexíveis, que atendem às necessidades de organizações digitais em busca de soluções robustas na nuvem Saraswat e Tripathi (2020).

3.2 Comparação de Preços sob Demanda entre as Principais Plataformas de Nuvem: *AWS*, *Azure* e *GCP*

A seleção das instâncias para hospedagem do aplicativo web cliente-servidor foi cuidadosamente embasada em critérios técnicos e econômicos que atendem às necessidades específicas do projeto.

A Tabela 1 apresenta as instâncias escolhidas, as quais foram selecionadas com o intuito de garantir que o aplicativo obtenha os recursos necessários para atender aos requisitos de desempenho e escalabilidade de maneira eficiente e econômica. A combinação dessas instâncias nas respectivas plataformas de nuvem proporciona uma abordagem flexível e versátil para a hospedagem do aplicativo, permitindo que a equipe de desenvolvimento escolha a opção que melhor se adapte aos requisitos particulares do projeto. Essa decisão estratégica visa otimizar o funcionamento do aplicativo e proporcionar uma experiência superior aos usuários, ao mesmo tempo em que otimiza os custos operacionais ao longo do ciclo de vida do projeto.

Os dados da Tabela 1 foram obtidos através dos sites oficiais das plataformas de nuvem: AWS (2023a), Azure (2023b) e GCP (2023a). Para garantir a precisão e confiabi-

Tabela 1 – Preços das instâncias

Tipo de Instância	**AWS**	**Azure**	**GCP**	**Preço AWS (por hora)**	**Preço Azure (por hora)**	**Preço GCP (por hora)**
Instância de servidor	t3.large	Standard_D2_v4	e2-standard-2	0,0832 USD	0.091 USD	0.07547 USD
Instância de Banco de dados	db.t3.micro	Standard_D2as_v4	db-f1-micro	0.022 USD	0.0960 USD	0,0112 USD

Fonte: Elaborado pelo autor.

lidade das informações, foram consultadas as páginas específicas que detalham os preços e especificações das instâncias disponíveis em cada plataforma.

A coleta de dados foi realizada com atenção às informações mais atualizadas disponíveis nos respectivos sites. É importante ressaltar que, devido à natureza dinâmica das plataformas de nuvem, os preços e especificações das instâncias podem ser atualizados periodicamente pelas provedoras de serviços. Portanto, os dados apresentados na Tabela 1 refletem a situação encontrada no momento da coleta, e é recomendado que sejam verificados e atualizados conforme necessário no momento da utilização em futuras análises ou projetos.

3.3 Aplicação baseada em Arquitetura de Microsserviços

A implementação de uma aplicação web cliente-servidor utilizando arquitetura de microsserviços pode oferecer algumas vantagens em termos de flexibilidade e desempenho. Nesse contexto, ao invés de adotar uma abordagem monolítica, onde todos os componentes estão integrados, a aplicação é dividida em microsserviços independentes. Cada microsserviço se encarrega de uma função específica, promovendo uma estrutura modular e mais fácil de ser mantida, segundo o RedHat (2023).

Com a arquitetura de microsserviços, a escalabilidade podeira ser mais eficiente. Diferentes partes da aplicação podem ser escaladas separadamente em resposta à demanda, evitando a necessidade de escalar a aplicação inteira. Isso não apenas melhora o desempenho, mas também otimiza o uso de recursos, reduzindo custos operacionais, de acordo com Amazon (2022c).

Na próxima seção, analisaremos diversos cenários de investigação para uma aplicação *web* cliente-servidor baseada em arquitetura de microsserviços. Esta abordagem visa ilustrar as diferentes formas como microsserviços podem ser integrados para criar uma aplicação web cliente-servidor. Ao apresentar esses cenários, procuraremos não apenas destacar as vantagens técnicas, mas também considerar implicações econômicas e estratégicas.

3.4 Cenários de Investigação

Nesta seção, exploramos uma variedade de cenários de investigação para uma aplicação web cliente-servidor baseada em microsserviços. Cada cenário apresenta uma abordagem distinta, desde a IaaS até estratégias multi-nuvem, com o objetivo de avaliar o custo da operação nas plataformas de nuvens selecionadas: AWS, Azure e GCP. O intuito é apresentar situações reais de uso da aplicação Web cliente-servidor baseada em microsserviços e analisar o impacto financeiro em diferentes configurações. Para conduzir essas implementações, procuramos manter ambientes virtuais o mais semelhantes possível, incluindo instâncias do *Ubuntu 22.04* e *MySQL 8.7*. No entanto, é importante ressaltar que cada plataforma de nuvem pode apresentar particularidades em suas configurações e preços. Portanto, embora busquemos manter os cenários comparáveis, é fundamental estar ciente de que a comparação ideal pode ser desafiadora devido às nuances específicas de cada nuvem.

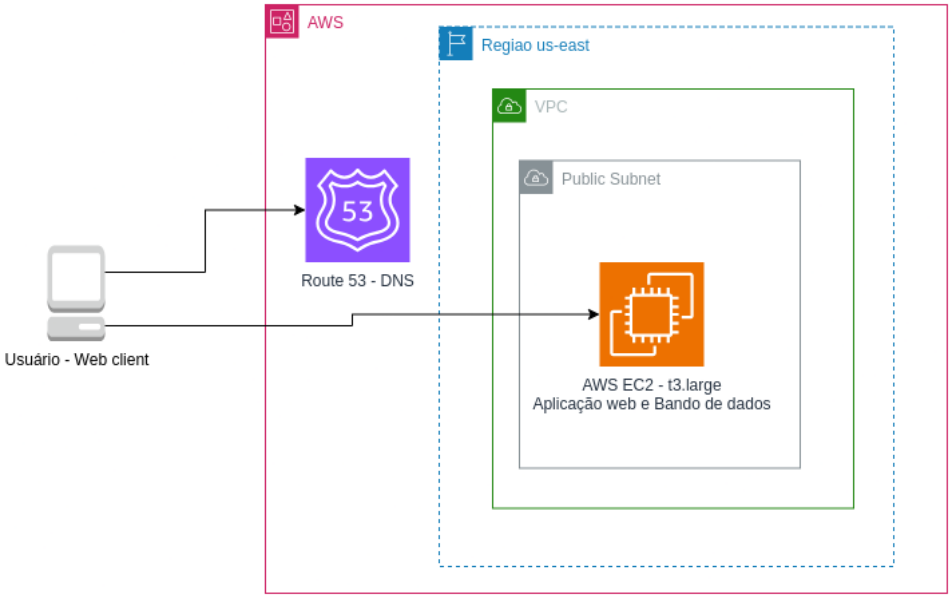
3.4.1 Cenários *IaaS*

No primeiro cenário (Figura 2), optamos por uma abordagem de IaaS na AWS. Nesse contexto, criamos uma única VM para a consolidação da investigação da aplicação *web* e do banco de dados SQL. A eficiência na comunicação entre esses componentes é alcançada por meio da virtualização de redes internas no sistema operacional da VM. Escolhemos utilizar instâncias *t3.large*, com um custo de 0,0832 *USD - United States dollar* (USD) por hora, de acordo com o fornecido no AWS (2023b). Isso resulta em um custo diário de 1,9968 USD e um custo mensal de 59,904 USD, considerando um mês com 30 dias. Além das instâncias, outros custos, como armazenamento, transferência de dados e recursos específicos da AWS, podem variar de acordo com o uso da aplicação.

Para acomodar o aumento na demanda, aprimoramos nossa implementação com duas VM (ver na Figura 3). Uma delas é dedicada à aplicação web, enquanto a outra é reservada para a instalação e configuração do banco de dados SQL. A comunicação entre essas VM ocorre por meio de uma rede local, otimizando significativamente a eficiência na troca de dados. Utilizamos duas instâncias *t3.large*, cada uma com um custo de 0,0832 USD por hora, segundo o fornecido por AWS (2023b). Isso resulta em um custo diário total de 3,9936 USD e um custo mensal de 119,808 USD, considerando um mês com 30 dias. Como no cenário anterior, os custos adicionais podem variar dependendo das necessidades específicas da aplicação.

Com a evolução da complexidade, introduzimos o uso de contêineres (Figura 4). Tanto a aplicação web quanto o banco de dados SQL são encapsulados em contêineres hospedados em serviços como o *Amazon - Elastic Container Service* (ECS) e o *Amazon Elastic Kubernetes Service* (EKS). Essa abordagem oferece maior flexibilidade e eficiên-

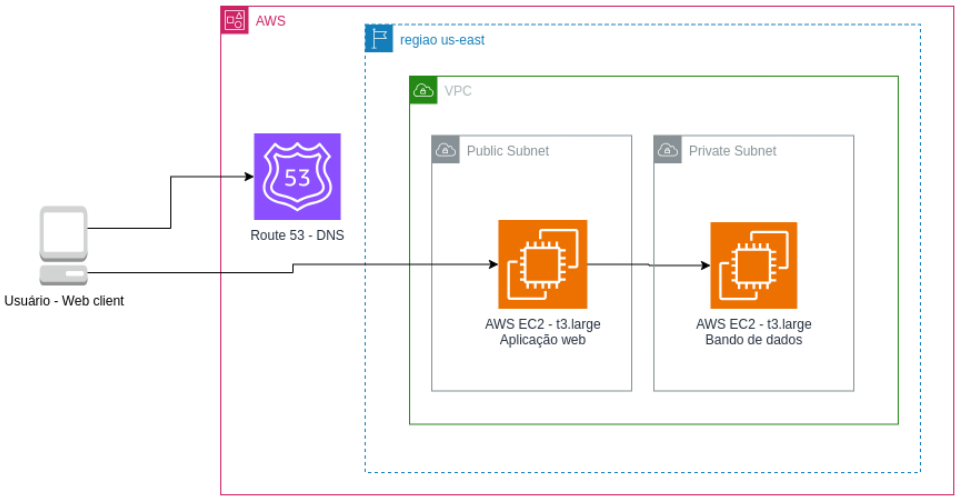
Figura 2 – Cenário 1



Fonte: Autoria própria.

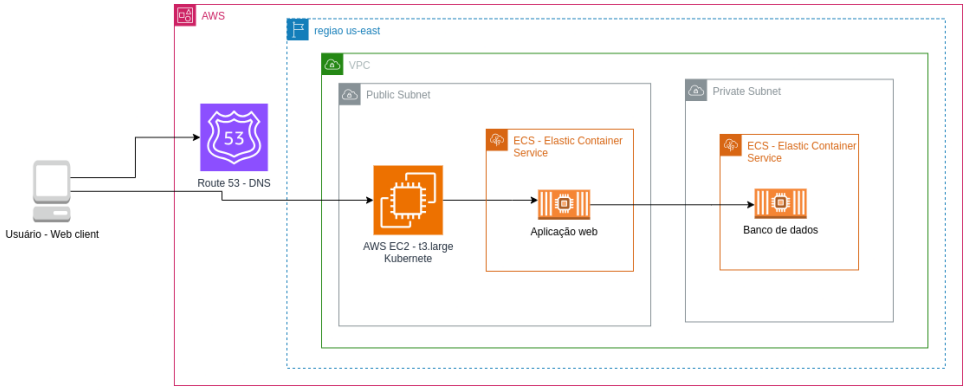
cia na gestão de recursos, permitindo uma implementação ágil e escalável. Esse cenário representa uma progressão significativa, incorporando um orquestrador de contêineres, o *Kubernetes*. Designamos uma VM para orquestrar e gerenciar os contêineres da aplicação web e do banco de dados SQL. A comunicação entre esses componentes é facilitada por meio da virtualização de redes, fornecendo uma infraestrutura dinâmica e escalável. Novamente, optamos por instâncias *t3.large* a 0,0832 USD por hora cada, conforme relatado por AWS (2023b), resultando em um custo diário de 1,9968 USD e um custo mensal de 59,904 USD, considerando um mês com 30 dias.

Figura 3 – Cenário 2



Fonte: Autoria própria.

Figura 4 – Cenário 3



Fonte: Autoria própria.

Em relação aos contêineres, é importante destacar que seu uso na AWS não implica em custos adicionais diretos. Você paga apenas pelos recursos das instâncias do *Amazon - Elastic Compute Cloud(EC2)* (EC2) onde os contêineres estão sendo executados. No entanto, é relevante notar que, embora os contêineres em si não gerem custos extras, é necessário ter um número mínimo de *hosts/workers* para orquestrar efetivamente os contêineres, de acordo com os dados de: AWS (2023b). Da mesma forma, a introdução do *Kubernetes* como orquestrador de contêineres não resulta em custos diretos adicionais. Entretanto, é necessário considerar a complexidade e o esforço necessários para configurar e gerenciar o *Kubernetes*.

Quanto aos volumes *Amazon - Elastic Block Store* (EBS), esses são usados para armazenamento persistente dos dados dos contêineres. Eles também podem representar um custo adicional, que varia de acordo com o tamanho e o desempenho dos volumes utilizados, acordo informado em AWS (2023b). Portanto, além de considerar os custos relacionados às instâncias do EC2 e aos serviços de orquestração de contêineres, é fundamental avaliar os custos associados aos volumes EBS, que podem variar dependendo das necessidades de armazenamento da aplicação.

3.4.2 Cenários *PaaS*

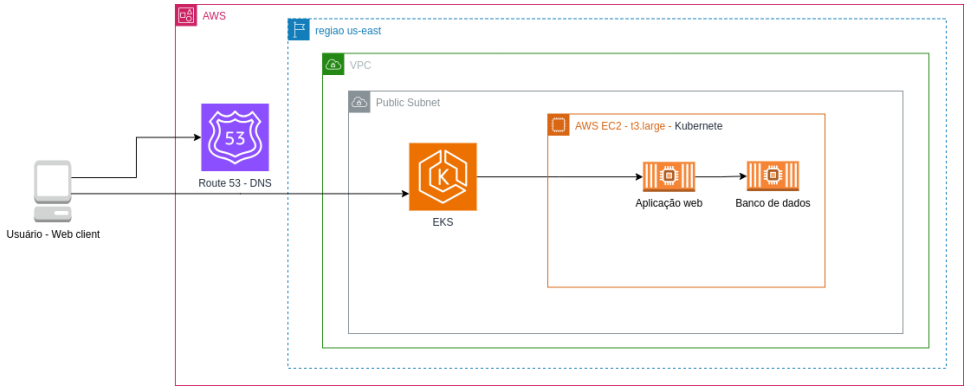
A transição para a PaaS na AWS marca um avanço significativo na evolução da investigação (Figura 5). Nesse contexto, a aplicação web é dividida em duas partes distintas: a parte estática, que utiliza HTTP puro e repousa no *Amazon - Simple Storage Service* (S3), e a parte dinâmica, que faz uso de *WebSockets* e permanece em execução em uma VM. Essa transformação traz consigo a adoção de uma variedade de serviços desempenhando funções essenciais, tais como a configuração do *API Gateway* e a implementação de *Virtual private network* (VPN).

Uma VM é alocada de forma exclusiva para a configuração da aplicação web, en-

quanto o serviço *Amazon - Relational Database Service* (RDS) assume a responsabilidade pelo gerenciamento do banco de dados *MySQL*. O armazenamento do conteúdo estático é fornecido pelo S3, demonstrando uma gestão simplificada e otimizada da infraestrutura.

No que se refere aos custos, é relevante mencionar que essa transição envolve três categorias principais: processamento, armazenamento e rede. Embora números específicos não sejam fornecidos, é crucial reconhecer a existência desses custos e que alguns deles são fixos, enquanto outros são variáveis. A flexibilidade crescente da infraestrutura à medida que evoluímos para a PaaS tende a tornar esses custos mais variáveis, dependendo da demanda.

Figura 5 – Cenário 4



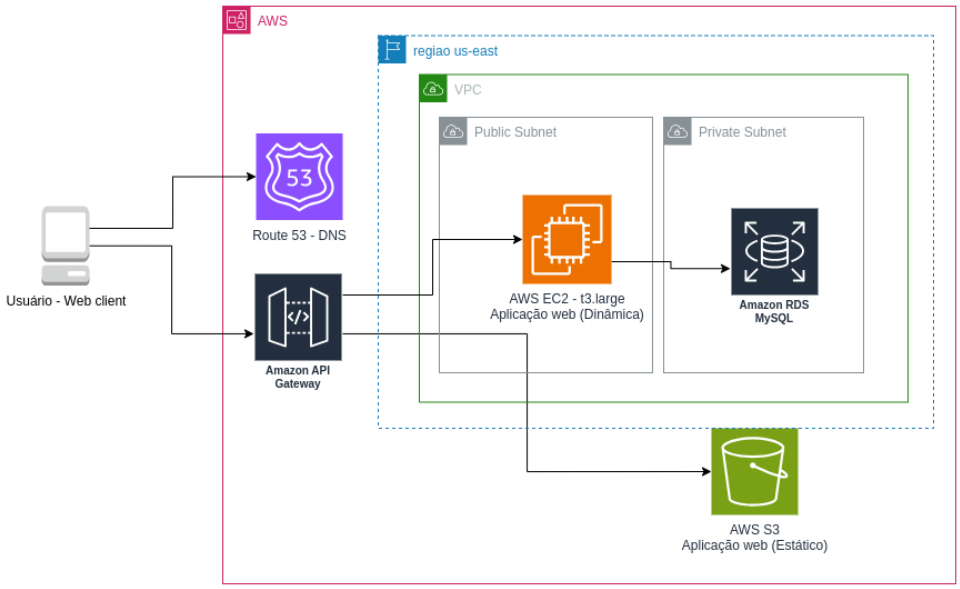
Fonte: Autoria própria.

Com a adoção desta abordagem avançada, que combina PaaS com o uso de contêineres, a aplicação web e o banco de dados SQL são encapsulados em contêineres gerenciados pelo ECS (ver na Figura 6). Essa estratégia oferece eficácia na orquestração e escalabilidade dos contêineres, proporcionando um ambiente ágil e simplificado para o desenvolvimento e operação da aplicação. É crucial destacar que, com a inclusão de contêineres e balanceadores de carga, também surgem custos associados que devem ser considerados.

No que tange ao orquestrador de contêineres, no cenário PaaS, é necessário pagar 0,10 USD por hora por cluster do EKS criado. Considerando um mês típico de 30 dias, isso resultaria em um custo diário de 2,4 USD. Além disso, os recursos da AWS usados para executar os nós do *Kubernetes*, como instâncias do EC2 e volumes do EBS, são faturados de acordo com o uso, o que pode variar dependendo da demanda.

Essas análises de custos e configurações ressaltam a necessidade de uma escolha estratégica baseada nas necessidades do aplicativo ao decidir entre cenários IaaS e PaaS. Cada cenário tem suas vantagens e desafios, bem como implicações financeiras, que devem ser avaliadas cuidadosamente no contexto de cada projeto. É fundamental considerar não apenas os custos diretos, mas também o potencial de otimização, escalabilidade e

Figura 6 – Cenário 5



Fonte: Autoria própria.

flexibilidade que cada cenário oferece.

3.4.3 Multinuvem

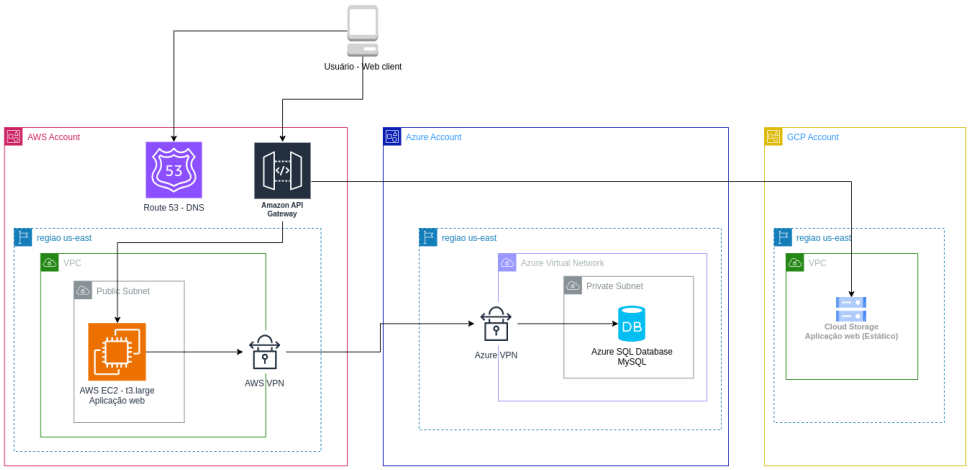
Neste cenário (Figura 7), exploramos a investigação em multinuvem, na qual distribuímos nossos microsserviços em várias plataformas de nuvem, incluindo AWS, Azure e GCP. A ênfase aqui recai fortemente sobre a segurança dos dados, com serviços isolados e comunicações protegidas por criptografia. Inicialmente, a comunicação entre esses serviços ocorre por meio de uma configuração de VPN manual entre os ambientes das diferentes nuvens, uma abordagem que, além de sua funcionalidade, pode ser considerada como uma etapa inicial para garantir a segurança dos dados. No entanto, essa abordagem gera *endpoints* para conexão e envolve vários passos para alcançar a solução final. Isso pode ser simplificado ou uniformizado com a implementação de criptografia em cada aplicação ou o uso de uma VPN, proporcionando um nível adicional de segurança aos dados em trânsito.

É crucial destacar que, durante a transição para a PaaS, vários serviços adicionais, como contêineres e balanceadores de carga, podem ser introduzidos. Esses serviços desempenham funções essenciais no ambiente multinuvem, mas é importante observar que sua utilização também pode gerar custos adicionais, uma vez que podem ser tarifados de acordo com o uso.

O cenário que apresentamos é uma ilustração de uma abordagem eficaz para a implementação em multinuvem. Nesse contexto, a aplicação é distribuída entre nuvens públicas, oferecendo flexibilidade, escalabilidade e otimização de custos. No entanto, é

importante destacar que cada plataforma de nuvem pode ter particularidades em suas configurações e preços. Portanto, ao tomar decisões, é essencial analisar com cuidado as necessidades específicas de sua aplicação.

Figura 7 – Cenário 6



Fonte: Autoria própria.

Nesse cenário multinuvem, é fundamental considerar os custos de manter recursos em diferentes nuvens, bem como as tarifas de transferência de dados entre as nuvens, que podem representar uma parcela significativa dos custos em alguns casos. Portanto, a análise e o planejamento criterioso dos custos desempenham um papel crucial na adoção de uma estratégia multinuvem com um *service mesh*. Além disso, vale ressaltar que a utilização de um *service mesh* pode simplificar a comunicação entre os microsserviços distribuídos nas diferentes nuvens, contribuindo para otimizar as operações e, ao mesmo tempo, adicionar eficiência aos custos.

3.4.3.1 *Service Mesh*

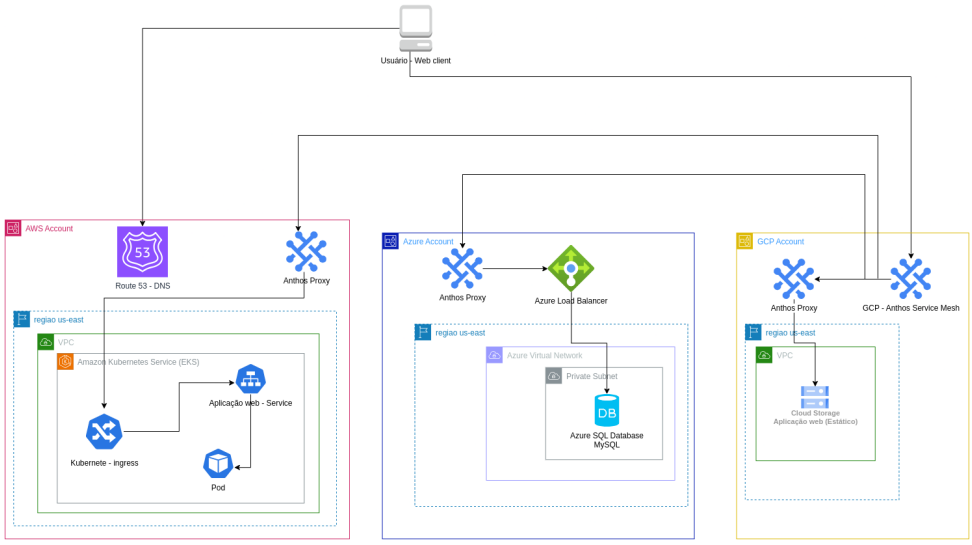
A introdução da malha de serviços (*service mesh*) ocorre como um avanço significativo nesse cenário de multinuvem. A malha de serviços simplifica e otimiza a comunicação entre os microsserviços, permitindo uma conexão mais eficiente e segura. Ela oferece uma solução de descoberta de serviços que simplifica a conexão entre os serviços distribuídos em diferentes nuvens, eliminando a necessidade de endereços (*endpoints*) complexos e configurações manuais.

Essa abordagem avançada de distribuição em multinuvem (ver na Figura 8), combinada com o uso da malha de serviços, visa a otimizar custos e desempenho. É importante observar que a adoção de serviços como contêineres e balanceadores de carga, embora forneça eficiência, também pode acarretar custos adicionais que devem ser considerados.

Em relação aos custos, é importante destacar que eles podem variar significativamente, dependendo das plataformas de nuvem utilizadas e da quantidade de tráfego.

O uso de uma malha de serviços pode adicionar complexidade e custos operacionais. No caso da AWS, é relevante notar que não há custos adicionais pelo uso do *AWS App Mesh* Amazon (2022a). Você paga somente pelos recursos da AWS consumidos pelo *proxy* implantado com os contêineres. Quanto ao *Azure Service Fabric* Azure (2023a), a cobrança é determinada pelas *APIs* do GCP ativadas em seu projeto. Se você desejar usar o *Anthos Service Mesh* GCP (2023b) como um serviço independente, evite ativar a *API Anthos* em seu projeto. Caso a intenção seja implementar o *Anthos Service Mesh* localmente ou em outras nuvens, será necessário assinar o *Anthos*.

Figura 8 – Cenário 7



Fonte: Autoria própria.

4 CONCLUSÕES

Conforme proposto no objetivo geral do trabalho, esta pesquisa buscou apresentar conceitos relacionados à implementação de uma aplicação *web* cliente-servidor nas principais plataformas de nuvem, a saber: AWS, Azure e GCP. A análise abordou desde a escolha de provedores de serviços em nuvem até a avaliação de cenários de implementação, com foco especial na eficiência e economia de custos. Ao resumir os resultados e contribuições, ressaltamos a importância de considerar não apenas os aspectos técnicos, mas também os elementos econômicos na escolha das instâncias de hospedagem, incluindo custos associados à expertise de mão de obra especializada.

O estudo teórico destacou algumas características dos principais provedores de serviços em nuvem. A AWS, Azure e GCP foram escolhidas como pontos de referência para a análise comparativa de preços, focando em serviços comuns a essas plataformas que apresentam pequenas variações em custos e características. A evolução dos cenários de implementação, desde a oferta de IaaS até PaaS, ilustrou a crescente complexidade e as oportunidades de otimização na gestão da aplicação.

Os cenários de investigação explorados proporcionaram uma visão abrangente, desde a alocação de máquinas virtuais até a distribuição da aplicação em ambientes multinuvem. A introdução da malha de serviços no contexto de multinuvem destacou-se como uma estratégia avançada para otimizar a comunicação entre microsserviços, ressaltando a importância da segurança dos dados em cenários distribuídos.

Ao explorar os distintos cenários, de IaaS e PaaS a multinuvem, analisamos as transformações técnicas. Desde a fase inicial, caracterizada pela implementação de máquinas virtuais, até o avançado estágio da introdução de serviços gerenciados e a complexa distribuição em multinuvem, que envolve uma malha de serviços, evidenciamos uma inovação significativa no panorama da computação em nuvem. Esse percurso, que inclui a ascensão de tecnologias como contêineres *serverless*, destaca não apenas a evolução técnica, mas também a adaptação à soluções mais ágeis.

É importante ressaltar que a dinâmica entre custos fixos ou variáveis, previsíveis ou imprevisíveis, varia em cada cenário. Os custos fixos podem ser mais proeminentes em modelos tradicionais, como o IaaS, enquanto modelos avançados, como a multinuvem, podem apresentar uma amplitude mais ampla de custos variáveis e desafios de previsibilidade. Portanto, ao considerar a implementação de uma estratégia específica, a compreensão profunda dessas nuances financeiras se torna ainda mais crucial. Isso não apenas influenciará a tomada de decisões no presente, mas também moldará a abordagem para otimização de custos e eficiência operacional no futuro.

Em termos de contribuições práticas, este trabalho oferece perspectivas para profissionais e acadêmicos, servindo como guia para a implementação de aplicações web em nuvem. As análises de custos, os diferentes cenários de implementação e a introdução da malha de serviços fornecem informações relevantes para a tomada de decisões informadas.

Reconhecemos, no entanto, que a evolução constante das plataformas de nuvem implica a adesão a ferramentas específicas, simplificando as operações, mas potencialmente limitando as mudanças de cenário. Embora isso possa significar uma transição de custos fixos para uma tendência descendente mais flexível, também introduz uma maior variabilidade suscetível a flutuações significativas em resposta à demanda de tráfego e processamento. Além disso, a implementação avançada exige um entendimento mais profundo das diferenças das nuvens utilizadas, o que implica uma equipe mais especializada.

Além disso, a dinâmica das plataformas de nuvem exige atualizações constantes nas análises de custos e comparações. No caso da implementação em multinuvem, embora promissora, ela demanda estudos mais aprofundados sobre segurança e gerenciamento. Esses são aspectos relevantes que merecem atenção em trabalhos futuros, visando aprimorar ainda mais a eficiência e segurança na implementação de aplicativos em nuvem. Vale destacar que os custos finais podem não ser tão evidentes no cálculo final, dada a natureza sob demanda de vários serviços oferecidos pelos provedores.

4.1 Trabalhos futuros

Para trabalhos futuros, recomendam-se as seguintes iniciativas:

- Uma análise mais aprofundada dos custos poderia incluir a consideração de fatores adicionais, como serviços de *big data* e aprendizado de máquina. Isso nos permitiria entender como esses elementos impactam os custos e como podem ser otimizados para alcançar um equilíbrio entre desempenho e economia.
- Explorar técnicas avançadas de criptografia para garantir a segurança dos dados em trânsito e em repouso. Implementar políticas de privacidade mais robustas e conformidade com regulamentos específicos.
- Investigar estratégias adicionais de escalabilidade, como a implementação de técnicas de balanceamento de carga mais avançadas. Explorar arquiteturas de microsserviços mais complexas para lidar com cargas de trabalho ainda maiores.
- Realizar estudos de caso em empresas que adotaram arquiteturas de microsserviços em multinuvem seria uma maneira de implementar os cenários de investigação. Isso nos permitirá entender como essas estratégias funcionam em ambientes do mundo real e identificar melhores práticas com base em exemplos reais.

REFERÊNCIAS

AMAZON. *AWS App Mesh*: Amazon web services. 2022. Disponível em: https://aws.amazon.com/app-mesh/. Acesso em: 19 set. 2023. 36

AMAZON. *Gartner Cloud Infrastructure Platform Services (CIPS)*: Amazon web services. 2022. Disponível em: https://aws.amazon.com/pt/blogs/aws/aws-named-as-a-leader-in-the-2022-gartner-cloud-infrastructure-platform-services-cips-magic-quadr Acesso em: 1 jul. 2023. 27

AMAZON. *What are Microservices?*: Microservices. 2022. Disponível em: https://aws.amazon.com/microservices/?nc1=h_ls. Acesso em: 10 jan. 2024. 29

ANALYTICS, A. *An Introduction to Serverless Computing and Function-as-a-Service*. 2021. Disponível em: https://www.advancinganalytics.co.uk/blog/2021/9/2/an-introduction-to-serverless-computing-and-function-as-a-service. Acesso em: 2022-11-18. 17

AWS. *Amazon EC2 pricing*: Amazon web services. 2023. Disponível em: https://aws.amazon.com/ec2/pricing/. Acesso em: 1 jul. 2023. 28

AWS. *Amazon EC2 pricing - On-demand*: Amazon web services. 2023. Disponível em: https://aws.amazon.com/pt/ec2/pricing/on-demand/. Acesso em: 1 jul. 2023. 30, 31, 32

AZURE. *Azure Service Fabric*: Azure cloud platform. 2023. Disponível em: https://learn.microsoft.com/en-us/azure/service-fabric/service-fabric-overview. Acesso em: 19 set. 2023. 36

AZURE. *Linux Virtual Machines Pricing*: Azure cloud platform. 2023. Disponível em: https://azure.microsoft.com/en-us/pricing/details/virtual-machines/. Acesso em: 1 jul. 2023. 28

BABU, S. A. et al. System performance evaluation of para virtualization, container virtualization, and full virtualization using xen, openvz, and xenserver. p. 247–250, 2014. 22

BERNSTEIN, D. Containers and cloud: From lxc to docker to kubernetes. *IEEE cloud computing*, IEEE, v. 1, n. 3, p. 81–84, 2014. 23

CHELLAPPA, R.; BARUA, A.; WHINSTON, A. Electronic infrastructure for a virtual university. *Communications of the ACM*, v. 40, n. 9, p. 56–58, 1997. ISSN 0001-0782. Disponível em: http://search.proquest.com/docview/26549524/. 13

Cloud Google. *What are containers*. 2022. Disponível em: https://cloud.google.com/learn/what-are-containers. Acesso em: 01 de outubro 2022. 22

GCP. *Compute Engine pricing*: Google cloud platform. 2023. Disponível em: https://cloud.google.com/compute/pricing. Acesso em: 1 jul. 2023. 28

GCP. *Google Cloud Platform*: Service mesh. 2023. Disponível em: https://cloud.google.com/anthos/service-mesh. Acesso em: 19 set. 2023. 36

HAMEL, G.; PRAHALAD, C. K. Competing in the new economy: Managing out of bounds. *Strategic Management Journal*, John Wiley & Sons, Ltd., Chichester, v. 17, n. 3, p. 237–242, 1996. ISSN 0143-2095. 13

JAIKAR, A. et al. Performance analysis of nas and san storage for scientific workflow. In: IEEE. *2016 International Conference on Platform Technology and Service (PlatCon)*. [S.l.], 2016. p. 1–4. 21

KAUSHIK, P. et al. Cloud computing and comparison based on service and performance between amazon aws, microsoft azure, and google cloud. In: IEEE. *2021 International Conference on Technological Advancements and Innovations (ICTAI)*. [S.l.], 2021. p. 268–273. 27

KUBERNETES, T. *What is Kubernetes*: Kubernetes. 2019. Disponível em: https://kubernetes.io/docs/concepts/overview/what-is-kubernetes/. Acesso em: 1 jul. 2023. 23, 24

MCCARTHY, J. Recursive functions of symbolic expressions and their computation by machine, part i. *Communications of the ACM*, ACM, v. 3, n. 4, p. 184–195, 1960. ISSN 00010782. 13

MUJEZINOVIĆ, A.; LJUBOVIĆ, V. Serverless architecture for workflow scheduling with unconstrained execution environment. In: IEEE. *2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. [S.l.], 2019. p. 242–246. 25

NIST. *The NIST Definition of Cloud Computing*. 2011. Disponível em: https://csrc.nist.gov/publications/detail/sp/800-145/final. Acesso em: 2022-09-07. 16, 18, 19

REDHAT. *What is a microservice?*: Microservices. 2023. Disponível em: https://www.redhat.com/en/topics/microservices/what-are-microservices. Acesso em: 10 jan. 2024. 29

RICHARDS, M. Software architecture patterns. In: O'REILLY MEDIA, INC. *Understanding Common Architecture Patterns and When to Use em*. [S.l.], 2017. p. 27–34. 14

SARASWAT, M.; TRIPATHI, R. Cloud computing: Comparison and analysis of cloud service providers-aws, microsoft and google. In: IEEE. *2020 9th international conference system modeling and advancement in research trends (SMART)*. [S.l.], 2020. p. 281–285. 27, 28