

INSTITUTO FEDERAL

Sertão Pernambucano

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO
SERTÃO PERNAMBUCANO
COORDENAÇÃO DO CURSO DE SISTEMAS PARA INTERNET
SISTEMA PARA INTERNET**

LUIS CARLOS AQUINO BARROS

**ANÁLISE DAS PLATAFORMAS PARA IMPLEMENTAÇÃO DA CULTURA
DEVOPS: UMA VISÃO GERAL DOS PRINCIPAIS PRESTADORES DE
SERVIÇOS EM NUVEM**

SALGUEIRO

2022

Dados Internacionais de Catalogação na Publicação (CIP)

B277 Barros, Luis Carlos Aquino.

Análise das plataformas para implementação da cultura DEVOPS: uma visão geral dos principais prestadores de serviços em nuvem / Luis Carlos Aquino Barros. - Salgueiro, 2022.
20 f. : il.

Trabalho de Conclusão de Curso (Sistemas para Internet) -Instituto Federal de Educação, Ciência e Tecnologia do Sertão Pernambucano, Campus Salgueiro, 2022.
Orientação: Prof. Esp. Francisco Junio da Silva Fernandes.

1. Desenvolvimento de software. 2. Cultura DevOps. 3. Serviços em nuvem. I. Título.

CDD 005.2

LUIS CARLOS AQUINO BARROS

ANÁLISE DAS PLATAFORMAS PARA IMPLEMENTAÇÃO DA CULTURA
DEVOPS: UMA VISÃO GERAL DOS PRINCIPAIS PRESTADORES DE
SERVIÇOS EM NUVEM

Trabalho de Conclusão de Curso
apresentado a Coordenação do
curso de Sistemas para Internet do
Instituto Federal de Educação,
Ciência e Tecnologia do Sertão
Pernambucano, campus Salgueiro,
como requisito parcial à obtenção do
título de Tecnólogo em Sistemas
para Internet.

Aprovado em: 08/03/2022.

BANCA EXAMINADORA

Prof. Francisco Junio da Silva Fernandes **Orientador(a)**
IF Sertão PE – Campus Salgueiro

Prof. Leonardo Corsino Campello
IF Sertão PE – Campus Salgueiro

Prof. Francenila Rodrigues Junior Souza
IF Sertão PE – Campus Salgueiro

SALGUEIRO

2022

Análise das plataformas para implementação da cultura DevOps: Uma visão geral dos principais prestadores de serviços em nuvem

Luís Carlos A. Barros¹, Francisco Junio da S. Fernandes²

Instituto Federal de Educação Ciência e Tecnologia do Sertão Pernambucano
BR 232, Km 504, sentido Recife, Zona Rural, 56000-000 – Salgueiro/ PE – Brasil

luis.carlos@aluno.ifsertao-pe.edu.br¹,

francisco.fernandes@ifsertao-pe.edu.br²

***Abstract.** The need and demand for software has been demanding from companies fast and frequent deliveries, in which it is necessary to reconcile speed and quality when developing software. For this, software engineering has been proposing over the years practices that optimize the software development process. As an evolution of agile methods comes the DevOps culture, which aims to increase productivity and speed within teams with its integration and continuous delivery practices. DevOps had its potential increased with the popularization of cloud computing. Thus, this paper brings an analysis of three platforms for implementing the DevOps culture in companies, as well as presenting an overview of the main cloud service providers and their characteristics.*

***Resumo.** A necessidade e procura por softwares vem exigindo das empresas entregas rápidas e frequentes, no qual é preciso conciliar a rapidez e qualidade ao se desenvolver um software. Para isso, a engenharia de software vem propondo ao longo dos anos práticas que otimizam o processo de desenvolvimento de software. Como evolução dos métodos ágeis surge a cultura DevOps, que visa aumentar com suas práticas de integração e entrega contínua a produtividade e rapidez dentro das equipes. A DevOps teve seu potencial elevado com a popularização da computação em nuvem. Assim, este trabalho traz uma análise de três plataformas para implementação da cultura DevOps nas empresas, bem como apresenta uma visão geral dos principais prestadores de serviços em nuvem e suas características.*

1. Introdução

O avanço da tecnologia criou no mercado de software uma competitividade entre as empresas, onde a frequência e rapidez na entrega tornaram-se fundamentais para se obter destaque em meio a essa competição. É preciso atenção ao realizar qualquer processo de forma rápida, um pequeno descuido ao se desenvolver um software pode ocasionar o seu mau funcionamento, tendo como consequências a perda de tempo, dinheiro e da confiança dos seus usuários.

Para cumprir esse desafio de rapidez e de qualidade no desenvolvimento de software, a engenharia de software vem estudando ao longo dos anos práticas e comportamentos que influenciam diretamente neste processo. Segundo Sommerville

(2011), não existem técnicas e métodos universais na engenharia de software adequados a todos os sistemas e todas as empresas. Ao invés disso, um conjunto diverso de métodos e ferramentas de engenharia de software tem evoluído nos últimos 50 anos. Já a qualidade de um software contempla uma série de objetivos da construção de um software, conhecidos como requisitos não-funcionais, tais como extensibilidade, capacidade de manutenção, reutilização de código, desempenho, escalabilidade, usabilidade e confiabilidade (JR., 2010).

Nos últimos anos uma cultura de trabalho vem ganhando espaço dentro das empresas, a DevOps. Com objetivo de trazer mais rapidez e qualidade no desenvolvimento de software através da automatização de tarefas e aumento na frequência de deploys¹. O DevOps teve seu potencial elevado com o crescimento da computação em nuvem, uma vez que a nuvem trouxe a essa cultura mais praticidade no processo de desenvolvimento, tornando esse processo mais colaborativo e rápido, oferecendo recursos como gerenciamento de código fonte e implementação de testes automatizados, evitando assim a necessidade de testes em máquinas físicas.

Muitas empresas de sucesso na internet – como Google, Amazon, Netflix, Flickr, GitHub e Facebook – perceberam que a tecnologia pode ser usada a seu favor e que o atraso no deploy para produção significa atrasar sua habilidade de competir e se adaptar a mudanças no mercado. É comum que elas realizem dezenas ou até centenas de deploys por dia (SATO, 2013).

Sendo assim, este trabalho traz uma análise de três plataformas para implementação da cultura DevOps nas empresas e apresenta uma visão geral dos principais prestadores de serviços em nuvem e suas características. O trabalho está organizado da seguinte forma: a seção 2 conta com a revisão de literatura como fundamento ao entendimento do trabalho. A seção 3 traz toda a metodologia adotada para realização do trabalho. A seção 4 apresenta alguns pontos sobre os provedores de serviço em nuvem, alvos da pesquisa. As seções 5 e 6 apresentam os resultados e conclusões.

2. Referencial Teórico

2.1. Engenharia de Software

Em sua obra Engenharia de Software, Sommerville (2011, p.6) definiu Engenharia de Software como uma abordagem sistemática para a produção de software; ela analisa questões práticas de custo, prazo e confiança, assim como as necessidades dos clientes e produtores do software.

¹ Fazer deploy de um sistema significa colocá-lo em produção.

Segundo o Boehm (2006), a engenharia de software surgiu a partir da década de 60, quando percebeu-se que modificar um software era muito mais fácil do que modificar um hardware e sua distribuição entre as máquinas não exigia altos custos. Com a expansão dos softwares e aumento da sua procura alguns problemas começaram a surgir, como código de difícil manutenção, elevado tempo de desenvolvimento, dentre outros problemas. Pensando em uma solução, o Comitê Científico da OTAN convoca duas conferências históricas de “Engenharia de Software” em 1968 e 1969. Essas conferências forneceram uma base sólida de compreensão do estado da engenharia de software, da prática que as organizações industriais e governamentais poderiam usar como base para determinar e desenvolver melhorias. Ficou clara a necessidade de métodos mais organizados e práticas mais disciplinadas, para escalar os projetos e produtos cada vez maiores que estavam sendo encomendados. (BOEHM, 2006, Traduzido)

A partir deste momento, pesquisadores e profissionais da área começaram a buscar métodos que padronizassem o desenvolvimento de softwares e os tornassem mais ágeis. O final da década de 1990 viu o surgimento de vários métodos ágeis, como: Desenvolvimento de Software Adaptativo, Crystal, Desenvolvimento de Sistemas Dinâmicos, eXtreme Programming (XP), Desenvolvimento Orientado a Recursos e Scrum (BOEHM, 2006, Traduzido).

Posteriormente surge o DevOps, uma evolução dos métodos ágeis citado pela primeira vez durante a conferência Velocity da O’Reilly em 2009, onde John Allspaw (Etsy.com) e Paul Hammond (Typekit) apresentaram uma proposta, com o objetivo de unir desenvolvedores (Dev) e administradores de infraestrutura de Tecnologia da Informação (Ops) de forma a promover a integração contínua até a entrega de software (VARIANI, 2016).

2.2. DevOps

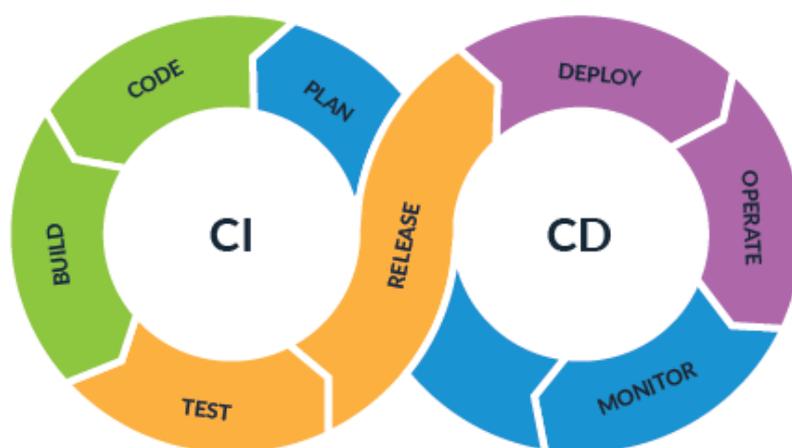
DevOps é um movimento cultural e profissional que tem como meta principal, promover uma melhoria na integração entre desenvolvedores e a equipe de infraestrutura. Ambas as funções são pressionadas pelas empresas, sendo desenvolvedores responsáveis por entregar valor em forma de funcionalidades e aplicações, enquanto a equipe de infraestrutura deve manter a estabilidade dos serviços. Com a sua implementação, as duas equipes passam a trabalhar juntas para melhorar tanto a produtividade dos desenvolvedores quanto a confiabilidade da equipe de infraestrutura. Essas duas equipes passam a participar do ciclo de vida completo de desenvolvimento e infraestrutura da aplicação, mantendo uma comunicação frequente de forma a aumentar a eficiência e melhorar a qualidade de serviços prestados aos clientes (BRANCO, 2020).

Além de uma mudança cultural, o movimento DevOps enfoca bastante nas práticas de automação das diversas atividades necessárias para tornar ágil as últimas

etapas do desenvolvimento e entregar código de qualidade em produção, como: compilação do código, testes automatizados, empacotamento, criação de ambientes para teste ou produção, configuração da infraestrutura, migração de dados, monitoramento, agregamento de logs e métricas, auditoria, segurança, desempenho, deploy (SATO, 2013).

Essa nova cultura se apresenta como solução para otimizar os processos de desenvolvimento e entregas de softwares. Para tanto aborda práticas em sua pipeline como integração, entrega, implantação e feedback, todos de forma contínua. A figura 1 apresentada a seguir, mostra um exemplo do pipeline² de um projeto com integração e entrega contínua.

Figura 1 - CI / CD Pipeline



Fonte: (Parasoft, 2021)

- **Integração Contínua (CI)**

Integração contínua é uma prática explicitamente identificada na metodologia de Extreme Programming (XP), que consiste no processo de integrar continuamente o trabalho desenvolvido em um repositório com os outros membros da equipe de desenvolvimento, executando os testes do trabalho integrado a cada check-in no repositório, permitindo assim detectar e localizar erros rapidamente (BRAGA, 2015).

- **Entrega Contínua (CD)**

Entrega Contínua é a prática que garante que o código compilado está pronto para ser disponibilizado nos diversos ambientes da organização. Depois de passar todos os controles de qualidade que visam garantir que o risco do deploy falhar é mais baixo (numa determinada máquina), pode então ser disponibilizada a nova alteração no ambiente em questão. Nesta prática, o código é disponibilizado depois do sucesso nos

² Pipeline de um projeto é uma série de etapas a serem realizadas para disponibilizar a nova versão de um software.

dois primeiros ambientes. Para produção existe um gatekeeper³ que decide se as alterações avançam para produção. Esta prática pressupõe obrigatoriamente o uso de um passo manual no processo (SOUSA, 2019).

- **Implantação Contínua**

A implantação contínua (CD) é o processo de implantar as mudanças no servidor de produção. Enquanto a entrega contínua se preocupa em deixar as mudanças testadas e validadas para serem implantadas no servidor de produção, a implantação contínua se preocupa em de fato instalá-las para produção. Quanto mais rápido esse processo for feito, mais rápido os desenvolvedores verão seu trabalho no ar e os usuários poderão utilizá-lo (PODGORNIK; GIBERTONI, 2020).

- **Feedback Contínuo**

O feedback contínuo pode ser resumido como o processo de transmitir para as pessoas retorno sobre seu desempenho, alinhado às expectativas do cargo ocupado de forma constante e consistente. A prática dele pode ser considerada uma das principais ferramentas para o gerenciamento de equipes, permitindo que os colaboradores entendam melhor suas responsabilidades dentro da equipe, e também passando mais segurança para que possam desempenhar sua função ao receberem a orientação necessária (BRANCO, 2020).

2.2.1. Microsserviços

Os microsserviços são definidos por Stoiber conforme citado por Lucio (2017, p28), como uma unidade de software autônoma que, juntamente com muitas outras, compõe uma grande aplicação. Ao dividir seu aplicativo em unidades pequenas, cada parte pode ser independentemente implantada e escalada.

Essa prática conciliada ao DevOps traz ainda mais rapidez as equipes. Ao se trabalhar com funcionalidades isoladas dentro do sistema é possível tratar as etapas da pipeline de forma individual, tornando a identificação e correção de erros mais rápidas sem que isso afete o sistema como um todo. Os microsserviços também proporcionam a possibilidade de se trabalhar com múltiplas equipes de forma simultânea, desse modo é possível garantir que cada funcionalidade esteja funcionando de forma correta, oferecendo assim uma melhor experiência ao usuário.

³ O Gatekeeper ou “Porteiro” é o responsável por certificar e aprovar todos os requisitos e entregas que a equipe de desenvolvimento produz.

2.3.Computação em Nuvem

A computação em nuvem ou Cloud Computing é um novo modelo de computação que permite ao usuário final acessar uma grande quantidade de aplicações e serviços em qualquer lugar e independentemente da plataforma, bastando para isso ter um terminal conectado à “nuvem” (PEDROSA; NOGUEIRA, 2011).

Com este novo modelo de virtualização de serviços, as empresas ganharam muito mais praticidade e segurança, uma vez que não faz mais necessário a presença de um centro de processamento de dados local, algo caro e que requer muito cuidado, já que todos os dados sensíveis da empresa eram alocados em um único lugar. Com a utilização do modelo em nuvem ganha-se flexibilidade, pois os dados e aplicações passam a ser alocados na nuvem por empresas que prestam este serviço. Dentre essas empresas, segundo pesquisa realizada por Bala et al., (2021) apresentada na figura 2, destacam-se: Amazon Web Services (AWS), a Microsoft Azure e o Google Cloud Plataforma (GCP).

Figura 2 - Quadrante Mágico para infraestrutura em nuvem e serviços de plataforma



Fonte: (Gartner, 2021)

Com base no quadrante mágico elaborado pela Gartner é possível identificar os principais prestadores de serviço em nuvem, assim teremos como alvo do trabalho os três principais, Amazon Web Services, Microsoft Azure e o Google Cloud Plataforma.

2.3.1. Amazon Web Services (AWS)

Lançada no mercado no ano de 2006 pela Amazon, a AWS pioneira em computação em nuvem, é amplamente confiável para grandes empresas e startups. Ela

ajuda as empresas a expandir e aumentar a competitividade, oferecendo infraestrutura como serviço (IaaS) sob demanda. Isso pode ser categorizado em banco de dados, serviços de computação, entrega e armazenamento de conteúdo e rede. Esses recursos podem ser utilizados separados ou em conjunto (NGQC, 2021).

A AWS tem a maior e mais dinâmica comunidade, com milhões de clientes ativos e dezenas de milhares de parceiros no mundo, além de ser projetada para ser um dos ambientes de computação em nuvem mais flexíveis e seguros atualmente disponíveis. A nuvem da AWS abrange 84 zonas de disponibilidade em 26 regiões geográficas em todo o mundo, com planos já divulgados para mais 24 zonas de disponibilidade e outras 8 regiões, através de vários data centers espalhados pelo mundo, conforme apresentado na figura 3, para que o usuário possa utilizar o mais próximo de sua localidade. Cada data center é instalado em uma zona de disponibilidade chamada Availability Zones (AZs), e no Brasil a zona mais próxima se localiza em São Paulo (AWS, 2022).

Figura 3 - Zonas e Regiões da AWS



Fonte: (Amazon, 2022)

2.3.2. Microsoft Azure (Azure)

Com mais de 200 produtos e serviços, o Microsoft Azure é uma plataforma de computação em nuvem que combina Infraestrutura como Serviço (IaaS), Plataforma como Serviço (PaaS) e Software como Serviço (SaaS) como suas ofertas (FASHAKIN, 2021). Segundo o Microsoft Azure (AZURE, 2022), com investimento de 1 bilhão de

dólares americanos em segurança contra ameaças cibernéticas, o Azure conquistou a confiança de 95% das empresas da Fortune 500⁴.

Quanto à presença global, segundo Fashakin (2021), o Azure tem a maior cobertura de data center para serviços em nuvem. Está disponível em mais de 54 regiões, abrangendo 140 países em todo o mundo. Ele associa regiões que estão no mesmo continente e que estão isoladas fisicamente umas das outras por pelo menos 480 km em conjunto de disponibilidade. Ele incentiva os usuários a arquitetar seus sistemas e aplicativos em torno dessas regiões, criando uma configuração de recuperação ativa/ativa, visando disponibilidade e isolamento (NASCIMENTO, 2020). A figura 4 mostra o mapeamento global do Azure.

Figura 4 - Zonas e Regiões do Azure.



Fonte: (Microsoft, 2022)

2.3.3. Google Cloud Plataform (GCP)

O Google Cloud Platform (GCP) é um serviço de computação em nuvem criado em 2008 pela Google, que oferece ferramentas e serviços que ajudam a desenvolver, implementar, entregar e gerenciar aplicações. Os serviços oferecidos pelo Google Cloud incluem computação, armazenamento, inteligência artificial (IA), e outros serviços de rede, desenvolvimento e gestão (BISONG, 2019).

Segundo a Google Cloud Plataform (GCP, 2022) sua nuvem está presente em 29 regiões, 88 zonas, 146 locais de borda e mais de 200 países e territórios, números esses

⁴ A Fortune 500 é uma lista anual compilada e publicada pela revista Fortune que contém as 500 maiores corporações dos Estados Unidos por receita total em seus respectivos anos fiscais.

que podem mudar uma vez que o google cloud continua em expansão em alguns territórios. A figura 5 mostra a essa distribuição no mapa.

Figura 5 - Zonas e Regiões do GCP



Fonte: (Google Cloud, 2022)

3. Metodologia

A metodologia abordada neste trabalho foi a pesquisa descritiva, no qual teve como base pesquisas bibliográficas. A pesquisa descritiva exige do investigador uma série de informações sobre o que deseja pesquisar. Esse tipo de estudo pretende descrever os fatos e fenômenos de determinada realidade (TRIVIÑOS, 2013).

Ainda foi utilizado o método dedutivo, uma concepção clássica que parte de princípios reconhecidos como verdadeiros e indiscutíveis e possibilita chegar a conclusões de maneira puramente formal, isto é, em virtude unicamente de sua lógica. Assim, o trabalho foi dividido em três etapas: Estudo bibliográfico, coleta de informações e análise dos dados coletados. A seguir será apresentado os métodos utilizados para realização desta pesquisa (GIL, 2008).

- Estudos Do Referencial Teórico
 - Estudo sobre o conceito de DevOps
 - Estudo sobre a evolução de engenharia de software
 - Estudo das Plataformas AWS, AZURE e GCP.
- Comparativo entre plataformas para o uso do DevOps.
- Obtenção de resultados após análise das comparações e referenciais teóricos.

3.1. DevOps e a Computação em Nuvem

Apesar de ser uma cultura, o DevOps necessita de ferramentas que auxiliem sua execução, para que assim seus princípios basilares sejam aplicados de forma eficiente. Em um passado recente, era comum as empresas possuírem data centers alocados em sua sede, o que possibilitava a implantação do DevOps. Entretanto a disponibilidade física é um ponto a se considerar ao adotar o uso de data centers locais, pois a empresa acaba por muitas vezes restringindo fisicamente o acesso de sua equipe, fazendo com que seja preciso estar presente na empresa para acessar funcionalidades a que necessitam diariamente, sendo o modelo de trabalho remoto uma opção inviável (NGOC, 2021).

Desse modo, a computação em nuvem trouxe ao DevOps mais praticidade, facilitando a comunicação e colaboração de toda a equipe sem a necessidade de encontros físicos, permitindo assim que as equipes sejam compostas por pessoas em diferentes lugares no mapa. A nuvem também traz a possibilidade de automatizar testes e executá-los em diversos ambientes, além de oferecer uma grande economia de tempo e redução de custos na prestação de serviços por demanda.

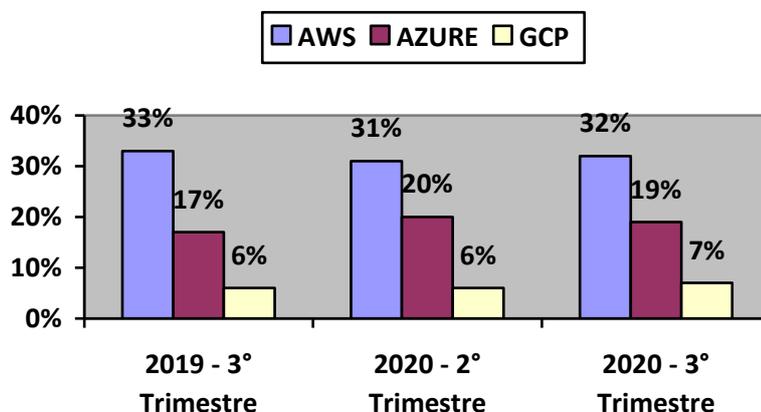
4. AWS vs. AZURE vs. GCP

Segundo Fashakin (2021), a escolha de um serviço em nuvem para DevOps pode variar entre custo, infraestrutura, localização de serviços de plataforma e setor. Assim o trabalho adotou como comparativo entre as plataformas a participação de mercado, custo e suporte nativo ao DevOps.

4.1. Participação no Mercado

A participação de mercado da AWS foi relatada com 32% no 3º trimestre de 2020, com uma receita de US \$36,5 bilhões, mais do que os outros provedores de serviços de nuvem apresentados no trabalho, combinados. O Azure detém uma participação de mercado de 19%, enquanto a participação de mercado do GCP é de 7% conforme apresentado no gráfico 1. (FASHAKIN, 2021)

Gráfico 1 - Participação no Mercado

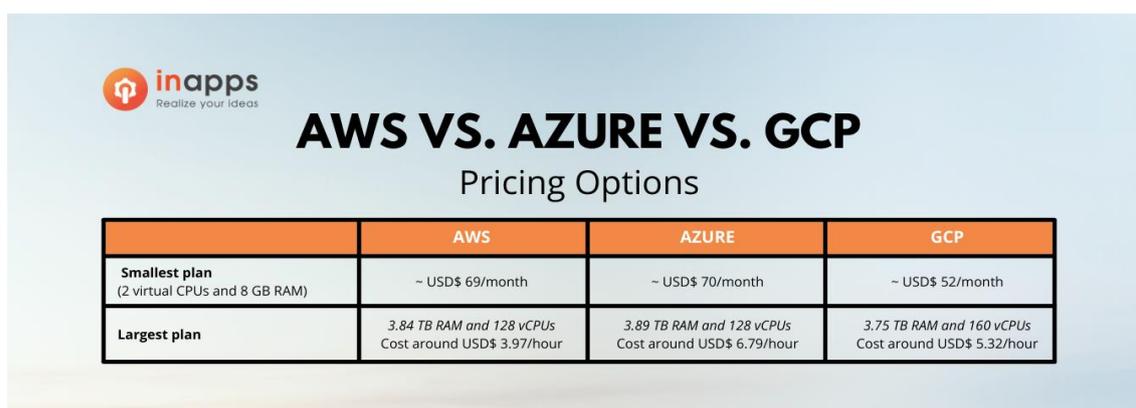


Fonte: (Canalys estimates, 2020)

4.2. Custo

Realizar um comparativo de preços ao se falar em provedores de serviços em nuvem é difícil, muitas variáveis devem ser consideradas, como planos disponibilizados, eventuais promoções, um bom gerenciamento do uso da nuvem em serviços por demanda, entre outras. Assim, a figura 6 traz um levantamento realizado pela Inapps Technology⁵, na qual apresenta uma média de custos.

Figura 6 - Comparativo de Custo



6

Fonte: (Inapps, 2021)

⁵ Empresa desenvolvedora de softwares com sede no Vietnã.

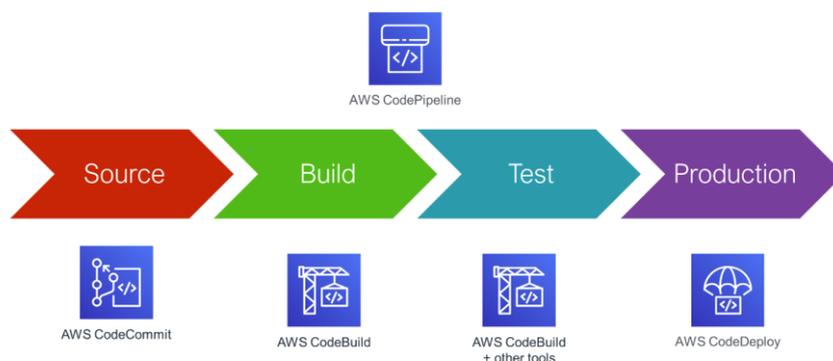
⁶ Na cotação atual do dólar americano, em março de 2022, 1 USD equivale a R\$ 5,26 reais.

4.3. Suporte Nativo ao DevOps

4.3.1. Amazon Web Services

A AWS possui em seu portfólio as seguintes ferramentas nativas que dão suporte à implementação da cultura DevOps. A figura 7 apresenta o modelo de pipeline com uso das ferramentas disponibilizadas na nuvem do AWS.

Figura 7 - AWS Pipeline CI/CD



Fonte: (K21Academy, 2021)

4.3.1.1. AWS CodePipeline

O AWS CodePipeline é um serviço de entrega contínua, que visa automatizar as fases de compilação, teste e implantação do processo de liberação sempre que ocorre uma mudança no código, permitindo testar facilmente todas as alterações de código e descobrir falhas enquanto elas ainda são pequenas e fáceis de corrigir. Assim é possível disponibilizar recursos e atualizações de forma rápida e confiável (AWS, 2022).

4.3.1.2. AWS CodeCommit

O AWS CodeCommit é um serviço de controle de código-fonte gerenciado, que disponibiliza um histórico de execuções de desenvolvimento de código, ajudando a resolver conflitos durante a mesclagem de contribuições de várias origens. O CodeCommit é seguro, altamente dimensionável e permite hospedar repositórios privados do Git⁷, ele permite que equipes colaborem com segurança no código, com contribuições criptografadas em trânsito e em repouso (AWS, 2022).

⁷ O Git é um sistema de controle de versões muito usado no desenvolvimento de software.

4.3.1.3.AWS CodeBuild

O AWS CodeBuild é um serviço de integração contínua totalmente gerenciado que compila o código-fonte, realiza testes e produz pacotes de software prontos para implantação, ele processa imediatamente cada compilação enviada e pode executar compilações distintas paralelamente, o que significa que elas não ficam esperando em uma fila (AWS, 2022).

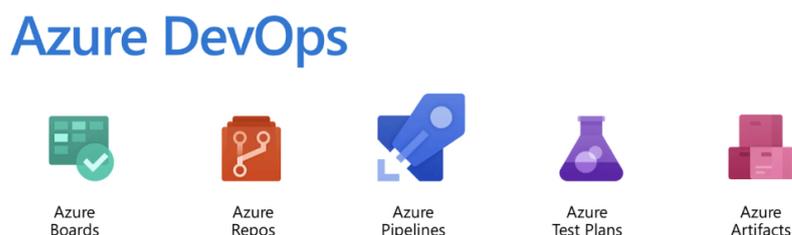
4.3.1.4. AWS CodeDeploy

O AWS CodeDeploy facilita o lançamento rápido de novos recursos, ajuda a evitar tempo de inatividade durante a implantação de aplicativos e lida com a complexidade de atualizá-los. Você pode usar o AWS CodeDeploy para automatizar implantações de software e eliminar a necessidade de operações manuais propensas a erros (AWS, 2022).

4.3.2. Microsoft Azure (AZURE)

O Azure possui em seu portfólio as seguintes ferramentas nativas para suporte à implementação da cultura DevOps. A figura 8 apresenta o modelo de pipeline com a nuvem do Azure.

Figura 8 - Ferramentas Azure DevOps



Fonte: (Stefanroth, 2019)

4.3.2.1. Azure Boards

O Azure Boards fornece às equipes de desenvolvimento de software, ferramentas interativas e personalizáveis necessárias para o gerenciamento de projetos de software. É um conjunto avançado de recursos, incluindo suporte nativo para processos ágeis, Scrum e Kanban, exibições de calendário, painéis configuráveis e relatórios integrados (AZURE, 2022).

4.3.2.2. Azure Repos

O Azure Repos é um conjunto de ferramentas de controle de versão que pode ser utilizado para gerenciar código. Ele mantém um histórico do desenvolvimento para que você possa revisar e até mesmo reverter para qualquer versão do código com facilidade,

sendo também muito útil para coordenar as alterações de código em sua equipe (AZURE, 2022).

4.3.2.3. Azure Pipelines

Combinando CI e CD, o Azure Pipelines cria e testa automaticamente projetos de código, para torná-los disponíveis para outras pessoas. Todas as alterações enviadas para o seu sistema de controle de código são automaticamente testadas e validadas, ele funciona com praticamente qualquer linguagem ou tipo de projeto. O uso do Pipelines ajuda a garantir um código consistente e de qualidade que esteja prontamente disponível para os usuários (AZURE, 2022).

4.3.2.4. Azure Test Plans

O Azure Test Plans é uma ferramenta utilizada para criar, analisar e examinar planos de testes em um projeto. É uma solução de gerenciamento de teste fácil de usar, baseada em navegador, fornece todos os recursos necessários para testes manuais planejados, teste de aceitação do usuário, testes exploratórios e coleta de comentários dos participantes. O Azure Test Plans é totalmente integrado ao Azure Pipelines para dar suporte ao teste dentro da integração contínua/implantação contínua (CI/CD) (AZURE, 2022).

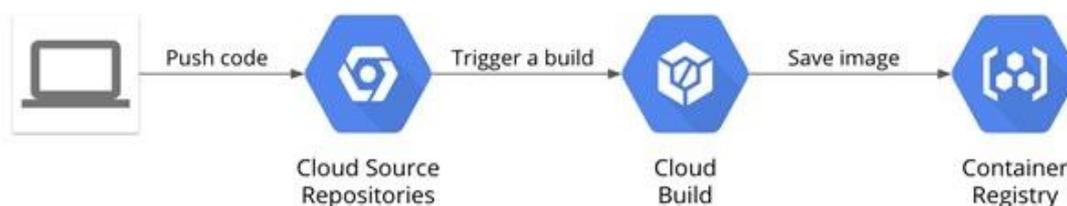
4.3.2.5. Azure Artifacts

O Azure Artifacts permite que os desenvolvedores compartilhem e consumam pacotes de diferentes feeds e registros públicos. Os pacotes podem ser compartilhados na mesma equipe, na mesma organização e até mesmo publicamente. Azure Artifacts dá suporte a vários tipos de pacotes, como NuGet, npm, Python, Maven e Universal Packages (AZURE, 2022).

4.3.3. Google Cloud Plataform(GCP)

O GCP possui em seu portfólio as seguintes ferramentas nativas para suporte à implementação da cultura DevOps. A figura 9 apresenta o modelo de pipeline com a nuvem do GCP.

Figura 9 - Ferramentas GCP DevOps



Fonte: (Dev Community, 2020)

4.3.3.1. Cloud Build

O Cloud Build é um serviço que executa suas versões na infraestrutura do Google Cloud Platform, importa o código-fonte do Google Cloud Storage, Cloud Source Repositories, GitHub ou Bitbucket⁸, ele executa um build de acordo com as especificações e produz artefatos como contêineres do Docker ou arquivos Java (GCP, 2022).

4.3.3.2. Cloud Deploy

O Google Cloud Deploy é um serviço gerenciado que automatiza a entrega de aplicativos para uma série de ambientes de destino em uma sequência de promoções definidas (GCP, 2022).

4.3.3.3. Artifact Registry

O Artifact Registry fornece um único local para gerenciar imagens de contêiner do Docker e pacotes. Ele se integra às ferramentas de CI/CD e aos ambientes de execução do Google Cloud para que você possa gerenciar todo o ciclo de vida do artefato (GCP, 2022).

4.3.3.4. Cloud Source Repositories

O Cloud Source Repositories é composto por repositórios Git particulares hospedados no Google Cloud Platform, que contam com pesquisa rápida de códigos indexados em todos os repositórios (GCP, 2022).

5. Resultados e Discussões

As três plataformas analisadas atendem as necessidades do DevOps, oferecendo ferramentas nativas que dão todo o suporte para as necessidades da cultura. Promovem de forma eficiente a construção de equipes mais colaborativas, códigos mais organizados com gerenciamento de histórico e muito mais rapidez ao lançar novas aplicações e atualizações no mercado devido a automatização de etapas fundamentais ao DevOps.

A AWS por possuir a maior parte da aderência do mercado acaba ganhando uma posição de destaque, entretanto o AZURE vem diminuindo a cada ano essa diferença de mercado. O GCP vem crescendo e conforme dados disponibilizados pela Flexera⁹ já vem alcançando números próximos aos seus concorrentes. A tomada de decisão entre qual nuvem escolher irá variar de acordo com as necessidades ou recursos que a empresa dispõe.

⁸ O github e o bitbucket são repositórios para armazenamento e controle de código fonte.

⁹ <https://www.flexera.com/blog/cloud/cloud-computing-trends-2021-state-of-the-cloud-report/>

6. Conclusões

A cultura DevOps não é algo que pode ser implementado de forma rápida, é preciso aceitação e colaboração por parte das equipes, uma vez que para seguir todos os princípios é necessário a participação direta dos membros em todo processo. Assim, empresas que já estão no mercado ao optarem pela migração à cultura DevOps, devem primeiramente treinar e remodelar suas equipes de modo a utilizarem metodologias ágeis e aos poucos aplicar os conceitos do DevOps até a implementação total.

7. Trabalhos Futuros

Como trabalhos futuros, pretende-se continuar aprofundando-se na cultura e seus ambientes de implementação.

- Analisar as melhores ferramentas baseadas em software livre para implementação DevOps.
- DevSecOps uma visão da cultura DevOps com mais segurança;
- Testar e Realizar um comparativo entre as ferramentas apresentadas no trabalho.

8. Referências

AWS. **Computação em Nuvem com a AWS**. 2022. Disponível em: <https://aws.amazon.com/pt/what-is-aws/>. Acesso em: 5 fev. 2022.

AZURE. **O que é o Azure?** 2022. Disponível em: <https://azure.microsoft.com/pt-br/overview/what-is-azure/>. Acesso em: 8 fev. 2022.

BALA, Raj; GILL, Bob; SMITH, Dennis; JI, Kevin; WRIGHT, David. Quadrante Mágico para infraestrutura em nuvem e serviços de plataforma. **Gartner**, [S. l.], 2021. Disponível em: <https://www.gartner.com/technology/media-products/reprints/AWS/1-271W1OT3-PTB.html>. Acesso em: 3 jan.2022.

BISONG, Ekaba. **Building Machine Learning and Deep Learning Models on Google Cloud Platform**. [s.l: s.n.]. DOI: 10.1007/978-1-4842-4470-8_29. Disponível em: https://link.springer.com/chapter/10.1007/978-1-4842-4470-8_2. Acesso em: 26 fev. 2022.

BOEHM, Barry. A View of 20th and 21st Century Software Engineering. **University of Southern California**, [S. l.], 2006. Disponível em: <https://dl.acm.org/doi/abs/10.1145/1134285.1134288>. Acesso em: 19 jan. 2022.

BRAGA, Filipe. UM PANORAMA SOBRE O USO DE PRÁTICAS DEVOPS NAS INDÚSTRIAS DE SOFTWARE. **UFPE**, [S. l.], p. 124, 2015. Disponível em: www.cin.ufpe.br/~posgraduacao. Acesso em: 21 fev. 2022.

BRANCO, Murilo Shinohata. Um estudo sobre DevOps. **Universidade Estadual de**

Londrina, [S. l.], p. 34, 2020. Disponível em: http://www.uel.br/cce/dc/wp-content/uploads/TCC_BANCA_MURILO_SHINOHATA_BRANCO.pdf. Acesso em: 7 jan. 2022.

FASHAKIN, Alexander. **Azure DevOps vs AWS DevOps vs GCP DevOps**. 2021. Disponível em: <https://devcontentops.io/post/2021/05/azure-devops-vs-aws-devops-vs-gcp-devops>. Acesso em: 12 jan. 2022.

GCP. **Locais do Cloud**. 2022. Disponível em: <https://cloud.google.com/about/locations>. Acesso em: 21 jan. 2022.

GIL, Antonio Carlos. Métodos e Técnicas de Pesquisa Social. **Annals of Ophthalmology**, São Paulo, v. 6, n. 1, p. 220, 2008. Disponível em: <https://ayanrafael.files.wordpress.com/2011/08/gil-a-c-mc3a9todos-e-tc3a9cnicas-de-pesquisa-social.pdf>. Acesso em: 1 mar. 2022.

JR., Hélio Engholm. **Engenharia de Software na Prática**. [s.l.: s.n.]. Disponível em: https://books.google.com.br/books?hl=pt-BR&lr=&id=Gq6LBgAAQBAJ&oi=fnd&pg=PA20&dq=engenharia+de+software&ots=ioHII4D5WV&sig=1_Ev2Mv3nzmTCfvnORuMHV9YKys#v=onepage&q=engenharia+de+software&f=false. Acesso em: 31 jan. 2022.

LUCIO, João Paulo Duarte. Análise comparativa entre arquitetura monolítica e de microsserviços. [S. l.], v. 110265, p. 110493, 2017. Disponível em: https://repositorio.ufsc.br/bitstream/handle/123456789/182309/TCC_corrigido.pdf?sequence=1&isAllowed=y. Acesso em: 18 mar. 2022.

NASCIMENTO, Gustavo Henrique Alvim. Computação em nuvem com a plataforma Microsoft Azure. **PUC Goiás**, [S. l.], 2020. Disponível em: https://repositorio.pucgoias.edu.br/jspui/bitstream/123456789/1070/1/TCC_Gustavo_Henrique_vers%C3%A3o_final.pdf. Acesso em: 14 fev. 2022.

NGOC, Nam Phuong Nguyễn. **DEVOPS AND CLOUD COMPUTING IN 2022: COMPARING AWS VS. GCP VS. AZURE**. 2021. Disponível em: https://www.inapps.net/devops-and-cloud-computing-in-2021-aws-vs-gcp-vs-azure/#Amazon_Web_Services. Acesso em 7 fev.2022.

PEDROSA, Paulo H. ...; NOGUEIRA, Tiago. Computação em Nuvem. **Unicamp**, [S. l.], p. 3–6, 2011. Disponível em: <http://www.ic.unicamp.br/~ducatte/mo401/1s2011/T2/Artigos/G04-095352-120531-t2.pdf>. Acesso em: 24 jan. 2022.

PODGORNIK, Hygor; GIBERTONI, Daniela. Estudo De Caso Sobre O Desenvolvimento De Uma Aplicação Scrum E Devops. **Revista Interface Tecnológica**, [S. l.], v. 17, n. 2, p. 244–255, 2020. DOI: 10.31510/infa.v17i2.978. Disponível em: <https://revista.fatectq.edu.br/index.php/interfacetecnologica/article/view/978/538>. Acesso em: 3 fev. 2022.

SATO, Danilo. **DevOps Na prática: entrega de software confiável e automatizada**. São Paulo. Disponível em: <https://www.casadocodigo.com.br/products/livro-devops>. Acesso em: 5 jan. 2022.

SOUSA, Leandro. DevOps - estudo de caso. **Instituto Politécnico de Coimbra**, [S. l.],

p. 100, 2019. Disponível em:
https://comum.rcaap.pt/bitstream/10400.26/31932/1/Leandro_Sousa.pdf. Acesso em: 8
jan. 2022.

TRIVIÑOS, Augusto Nivaldo Silva. **Introdução à Pesquisa em Ciências Sociais: a Pesquisa Qualitativa em Educação – O Positivismo, A Fenomenologia, O Marxismo**. [s.l: s.n.]. v. 1 DOI: 10.33081/formação.v1i20.2335. Disponível em:
https://edisciplinas.usp.br/pluginfile.php/4233509/mod_resource/content/0/Trivinos-Introducao-Pesquisa-em_Ciencias-Sociais.pdf. Acesso em: 15 fev. 2022.

VARIANI, Aiefe. **DEVOPS: INTRODUÇÃO, COMO SURTIU, CONCEITOS E FERRAMENTAS**. 2016. Disponível em:
<https://alefevariani.wordpress.com/2016/02/17/devops-introducao-como-surgiu-conceitos-e-ferramentas/>. Acesso em: 25 fev. 2022.