

Viabilidade da arquitetura em nuvem para instituições educacionais

Matheus Antonio Monteiro
Graduando em Engenharia de Software – Uni-FACEF
matheusantoniomonteiro035@gmail.com

Prof. Me. Leandro Borges
Mestre em desenvolvimento regional – Uni-FACEF
leandro.borges@aion.eng.br

Resumo

Nestes últimos dois anos vivemos um período no qual o ensino a distância se tornou realidade para todas as instituições de ensino devido a pandemia e como a grande maioria das instituições não se prepararam, a infraestrutura dos servidores de algumas instituições não estava devidamente configurada para ter a quantidade de acessos simultâneos que tiveram e acabaram ficando instáveis e até mesmo fora do ar prejudicando o acesso dos discentes e docentes nos primeiros dias de ensino *online*. As instituições educacionais passaram por esse problema nas primeiras semanas, acessar a plataforma *online* estava praticamente impossível devido à alta taxa de conexões simultâneas. Esse problema poderia ser facilmente contornado utilizando uma arquitetura em nuvem para a infraestrutura das instituições que se auto escalasse conforme a necessidade. Portanto esse artigo tem como objetivo apresentar e comprovar a viabilidade da utilização da arquitetura em nuvem para instituições educacionais independentes do seu escopo ou nível de ensino.

Palavras-chave: Amazon Web Service. Arquitetura em nuvem. Ensino *Online*. Computação em Nuvem.

Abstract

On the last two years we lived in a period which online learning became reality for all educational institutions due the pandemic, and because the server infrastructure of some institutions was not ready to support high simultaneous connections, they were unstable and even down harming students' and professors' access in the firsts three days of online learning. The institutions went through these problems, the platform was virtually impossible to access due the high simultaneous connections tax. This problem could be easily sorted out with cloud architecture for the servers' institutions infrastructure. Therefore, the goal of this article is to prove the feasibility of using cloud architecture by educational institutions regardless of scope of it.

Keywords: Amazon Web Service. Cloud Architecture. Online Learning. Cloud Computing.

1 Introdução

A computação em nuvem está cada vez mais frequente no dia a dia das pessoas, seja assistindo a um filme no Netflix ou jogando aquele jogo que te prende por horas à frente do celular. Há uma tendência cada vez maior na utilização da nuvem para armazenar as aplicações devido a suas vantagens, tais como disponibilidade *full time* (24/7), flexibilidade, facilidade de implementação, segurança e economia as vantagens mais buscadas com sua utilização.

A nuvem é um serviço que disponibiliza seus recursos, dedicados ou compartilhados, via internet (ROBU, 2012; YOO, 2011). Isso mostra a facilidade que é utilizar esse tipo de computação. Não há a necessidade de investir em *datacenters* enormes para armazenar suas aplicações, basta criar uma conta em um dos vários *players* disponíveis no mercado como Amazon AWS, Microsoft Azure entre outros e começar a usufruir de suas vantagens.

Os ganhos sugeridos pela computação em nuvem, com destaque para a disponibilidade, auto escalonamento, segurança e economia, e o aumento da educação a distância no último ano devido a pandemia do Corona Virus, mostraram como o uso desse tipo de arquitetura nos setores educacionais públicos e privados é necessário.

A proposta deste artigo é mostrar a viabilidade de se começar o quanto antes a utilização do serviço da computação em nuvem em entidades educacionais como escolas e faculdades, mostrando as vantagens de sua utilização focando em ambientes para educação à distância como um *moodle* e principalmente na segurança de acesso utilizando o provedor de nuvem da Amazon, a AWS, focando na autodisponibilidade das aplicações e na segurança da aplicação utilizando serviços da AWS que tem o seu foco em segurança. Será apresentado também uma arquitetura que poderá ser utilizada para criar o ambiente na prática e também duas estimativas de custos, mostrando uma estimativa de quanto se gastará por mês para manter o ambiente da arquitetura.

2 Referencial Teórico

2.1 Computação em nuvem

A computação em nuvem pode ser definida como

Um modelo que possibilita acesso, de modo conveniente e sob demanda, a um conjunto de recursos computacionais configuráveis (por exemplo, redes, servidores, armazenamento, aplicações e serviços) que podem ser rapidamente adquiridos e liberados com mínimo esforço gerencial ou interação com o provedor de serviços. (NIST, 2011, p. 02).

Segundo Neto (2015), a computação em nuvem permite que haja uma divisão dos processos relacionados ao negócio em TI e que isso faz com que utilizando a computação em nuvem você tem também uma maior elasticidade na infraestrutura da aplicação.

A AWS define computação em nuvem como “[...] a entrega de recursos de TI sob demanda por meio da internet com definição de preço de pagamento conforme o uso. [...]”. (AWS Cloud Computing, S.I., *online*)

Segundo a mesma plataforma (AWS, S.I.) com o uso da computação em nuvem não existe mais a necessidade de gastar recursos para montar um datacenter para sua empresa, basta apenas começar a utilizar a computação em nuvem e assim a companhia que optar pelo uso da arquitetura em nuvem só precisa gastar com o que vai utilizar gerando assim uma maior economia.

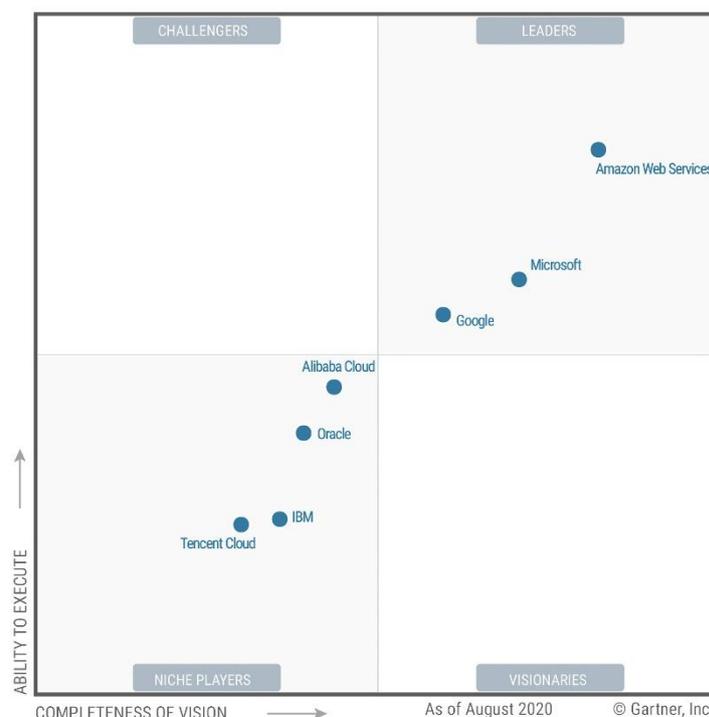
2.2 Amazon Web Service - AWS

A Amazon Web Services é uma das várias plataformas de nuvem que existem. Sendo ela a pioneira e a mais adotada hoje em dia. Ela oferece mais de 200 serviços completos, possui a maior rede de comunidade, é mais segura. (AWS, S.I.)

Segundo a plataforma (AWS, S.I., *online*), “Há mais de 14 anos, a AWS entrega serviços em nuvem para milhões de consumidores no mundo todo que executam uma grande variedade de casos de uso”. Até mesmo órgãos governamentais estão optando por colocar seus serviços na AWS. Tendo assim uma maior economia, mais agilidade e foco em inovação rápida.

A figura 1 mostra os provedores de nuvem mais utilizados de 2020, sendo a AWS a mais utilizada disparadamente.

Figura 1 Provedores de nuvem mais utilizados 2020



Fonte: AWS, 2020

2.3 Serviços AWS apresentados

Nessa seção vamos apresentar os serviços e conceitos que ainda serão mostrados na arquitetura presente na seção 3.

2.3.1 Virtual Private Cloud – VPC

“A *Amazon Virtual Private Cloud* (Amazon VPC) permite provisionar uma seção logicamente isolada na Nuvem AWS, na qual é possível executar recursos da AWS em uma rede virtual definida por você”. (AWSVPC, S.I., *online*).

A VPC tem semelhança com uma rede tradicional de um datacenter, mas “...com o benefício de usar a infraestrutura dimensionável da AWS.” (AWSVPC, S.I., *online*).

2.3.2 Subnets

Subnets são sub-redes que possuem intervalos de IP's dentro de uma VPC. Com a sua utilização nós podemos especificar se elas serão públicas, ou seja, outros hosts fora da VPC poderão acessar ela, ou privadas, na qual somente hosts que tiverem na rede interna da VPC conseguem acessar (AWSVPC, S.I.).

2.3.3 Internet Gateway

Segundo a documentação da AWS o *Internet Gateway* é “Um gateway que você anexa à VPC para permitir a comunicação entre recursos na VPC e a Internet.” (AWSVPC, S.I., *online*).

Com a criação de um *gateway* será possível fazer com que a VPC e todos os serviços que tiverem dentro dessa rede se comunique com a internet podendo buscar por atualizações, baixar alguma ferramenta nos servidores criados e se comunicar com hosts fora da AWS (AWSVPC, S.I.).

2.3.4 Grupos de segurança

“Um *security group* atua como um firewall virtual para sua instância para controlar o tráfego de entrada e saída. Quando você executa uma instância na VPC, é possível atribuir até cinco grupos de segurança à instância”. (AWSSecurityGroup, S.I., *online*)

Esses grupos são a principal forma de segurança dos servidores e por isso devem ser devidamente configurados liberando apenas as portas necessárias para quem realmente pode ter acesso a elas, caso contrário os servidores estarão vulneráveis a ataques.

Ainda segundo a documentação da AWS, um grupo de segurança funciona:

Para cada *security group*, adicione *regras* que controlam o tráfego de entrada para instâncias e um conjunto separado de regras que controlam o tráfego de saída. Esta seção descreve as informações básicas que você precisa saber sobre grupos de segurança para a VPC e suas regras. (AWSSecurityGroup, S.I., *online*)

2.3.5 Elastic Compute Cloud – EC2

“O *Amazon Elastic Compute Cloud - EC2* é um serviço da web que fornece capacidade de computação escalável (literalmente, serviços dos datacenters da Amazon) que pode utilizar para criar e hospedar os seus sistemas de software” (AWSEC2, S.I., *online*)

Segundo a própria AWS (AWSEC2, S.I.), ao fazer o uso do serviço de *elastic compute cloud*, não há a necessidade de investir em hardwares e isso faz com que o você foque mais tempo no desenvolvimento do que na configuração de

servidores físicos, além de não precisar ter uma sala específica com os cuidados necessários para armazenar as máquinas com o uso da AWS isso é de responsabilidade deles.

2.3.6 Bastion

Bastion é o nome dado a uma instância EC2 que tem um propósito especial de servir como primeiro acesso (AWSBastion, S.I.). Como na arquitetura que será apresentada os servidores não terão IP público será necessário ter um *bastion* para que o time de infraestrutura possa acessar esses servidores para uma possível atualização ou análise.

2.3.7 Elastic Load Balance

“O *Elastic Load Balancing* distribui automaticamente o tráfego de entrada de aplicações entre diversos destinos, como instâncias do Amazon EC2, contêineres, endereços IP, funções do Lambda e dispositivos virtuais” (AWSLoadBalance, S.I., *online*)

O ELB (*Elastic Load Balancing*) recebe as requisições dos usuários e direciona ele entre as instâncias EC2 que forem atribuídas a esse serviço, então caso tenha 5 instâncias rodando o mesmo projeto e elas estiverem atribuídas ao ELB ele distribuirá os usuários entre esses 5 servidores.

2.3.8 Elastic File System

O serviço de *Elastic File System* “[...] oferece armazenamento de arquivos para a instância do Amazon EC2 [...]” (AWSElasticFileSystem, S.I., *online*). Ele é usado como disco compartilhado, assim não é necessário criar volumes muito grandes caso o projeto tenha muitos arquivos de mídia e sua utilização é simples de ser aplicada, pois ele não necessita de um servidor (AWSElastFileSystem, S.I.).

2.3.9 Elastic IP address

Um endereço de IP elástico é um IP público do tipo IPV4 que não sofre alterações. Assim que você associa esse IP à uma instância EC2, ele não mudará até que você faça o *release* desse IP, ou seja, liberar ele (AWSElasticIP, S.I.). Isso facilitará o acesso ao time de infraestrutura no servidor *bastion*.

2.3.10 Auto Scaling Group

Um *auto scaling group* é um grupo de instâncias EC2 que servem como um agrupamento lógico para que o ambiente tenha uma escalabilidade automática e possa estar sempre disponível. (AWSASG, S.I.)

Segundo a documentação da AWS, *auto scaling group* pode ser definido como:

Um grupo do Auto Scaling também permite que você use recursos do Amazon EC2 Auto Scaling como substituições de verificação de integridade e políticas de escalabilidade. A manutenção do número

de instâncias em um grupo de Auto Scaling e a escalabilidade automática são os principais recursos do serviço Amazon EC2 Auto Scaling.(AWSASG, S.I., *online*)

2.3.11 Relational Database Service – RDS

O serviço de banco de dados relacional da AWS, o RDS, é uma *web service* que permite hospedar banco de dados relacionais na nuvem da AWS. Atualmente possui suporte para várias plataformas de banco de dados como MySQL, Oracle, PostgreSQL, MariaDB, Amazon Aurora e até mesmo Microsoft SQL Server. (AWSRDS, S.I.). Sua utilização facilita realizar ações no banco de dados e deixa a base de dados separada do servidor da aplicação melhorando assim a segurança e facilitando a manutenção dela.

2.3.12 Identity and access management – IAM

O *Identity and Access Management* é um serviço presente na AWS que nos permite realizar a segurança de acesso à conta através da criação de usuários e controlando a autorização desses usuários (AWSIAM, S.I.)

Após a criação da conta na plataforma a conta criada tem acesso root na AWS e segundo o guia do usuário da amazon é uma boa prática criar um usuário administrador para a conta e usar a conta root apenas para criar esse usuário e realizar o gerenciamento da conta, como métodos de pagamento e visualizar a fatura do mês. (AWSIAM – S.I.)

2.3.13 Amazon simple storage service – S3

O *Amazon Simple Storage Service* ou simplesmente S3, é um serviço de armazenamento presente na AWS o qual permite que haja o armazenamento de arquivos, muito utilizado para armazenar bakups de documentos. Esse serviço é composto por *buckets*, objetos e chaves (AWSS3, S.I.).

Segundo a documentação do S3, um *bucket* pode ser representado como:

[...] um contêiner para objetos armazenados no Amazon S3. Cada objeto está contido em um bucket. Por exemplo, se o objeto chamado *photos/puppy.jpg* estiver armazenado no bucket *awsexamplebucket1* da região Oeste dos EUA (Oregon), ele poderá ser endereçado usando o URL <https://awsexamplebucket1.s3.us-west-2.amazonaws.com/photos/puppy.jpg>. (AWSS3, S.I., *online*)

Segundo a mesma documentação os *buckets* precisam ter nomes únicos, já que o serviço é de escala global. Então caso formos criar um *bucket* ele não poderá ter o nome de *awsexamplebucket1*, pois esse nome já está sendo usado por outro usuário da AWS. (AWSS3, S.I.)

Já os objetos podem ser classificados como “[...] entidades fundamentais armazenadas no Amazon S3. [...]” uma vez que eles são os documentos propriamente dito, então caso um arquivo de texto seja armazenado no *bucket* esse arquivo é considerado um objeto. (AWSS3, S.I.)

“Os objetos consistem em metadados e dados de objeto. [...] Os metadados são um conjunto de pares de nomes e valor que descrevem o objeto” (AWSS3, S.I., *online*)

Falando sobre as chaves dos *buckets*, segundo a documentação da Amazon

Uma chave é um identificador exclusivo de um objeto em um bucket. Cada objeto em um bucket tem exatamente uma chave. A combinação de um bucket, uma chave e um ID de versão identificam exclusivamente cada objeto. Portanto, é possível pensar no Amazon S3 como um mapa de dados básico entre "bucket + chave + versão" e o objeto em si. (AWSS3, S.I., *online*)

Essas chaves são basicamente o nome dos objetos por exemplo “arquivo.pdf” é a chave do objeto, então não é possível ter o mesmo nome para arquivos diferentes no mesmo *bucket*. (AWSS3, S.I.)

2.3.14 *Cloudwatch*

O *cloudWatch* é um serviço que monitora os outros serviços e aplicações presentes na conta. Isso facilita que as pessoas consigam visualizar o que está acontecendo com a infraestrutura do projeto como uso de CPU, memória e até mesmo o valor gasto com essa infraestrutura (AWSCloudWatch, S.I.).

Segundo a AWS com a utilização do *CloudWatch* “Não é mais necessário configurar, gerenciar e dimensionar sua própria infraestrutura e sistemas de monitoramento” (AWSCloudWatch, S.I., *online*).

Você pode também criar alarmes e vincular um e-mail à conta aos alarmes e quando o nível de utilização de CPU estiver alto um e-mail será enviado alertando sobre o ocorrido, assim os responsáveis pela infraestrutura poderão tomar medidas antes que o servidor pare de funcionar (AWSCloudWatch, S.I.).

2.3.15 *Cloudwatch agent*

O *cloudWathc agent* permite que você consiga coletar métricas e logs de servidores, tanto EC2 quanto *on-demand*, que não são coletados por padrão no *cloudwatch*, independente do sistema operacional utilizado, Linux ou Windows (AWSCloudAgent, S.I.).

2.3.16 *Amazon certificate manager*

Segundo a documentação da AWS, “O *AWS Certificate Manager* (ACM) lida com a complexidade de criar, armazenar e renovar chaves e certificados SSL/TLS X.509 públicos e privados que protegem seus sites e aplicativos da AWS.” (AWSCertificateManager, S.I., *online*).

Como muitas fundações educacionais trabalham com dados sensíveis para acesso à plataforma é necessário que esses dados estejam criptografados e para isso é necessário registrar e utilizar um certificado SSL para a aplicação. O ACM facilita a criação e gerência desses certificados.

2.3.17 *Elasticache*

Segundo a documentação do serviço, *ElastiCache*:

O Amazon ElastiCache facilita a configuração, o gerenciamento e a escala de ambientes de cache de memória distribuídos na Nuvem AWS. Ele fornece um cache de memória econômico, de alta performance e redimensionável, ao mesmo tempo em que remove a complexidade associada à implantação e ao gerenciamento de um ambiente de cache distribuído. (AWS>ElastiCache, S.I., *online*)

Com a utilização do *elasticache* melhora a experiência do usuário e facilita a análise dos caches da aplicação, pois o serviço possui várias métricas que podem ser facilmente visualizadas em forma gráfica.

2.3.18 Reserva de instâncias

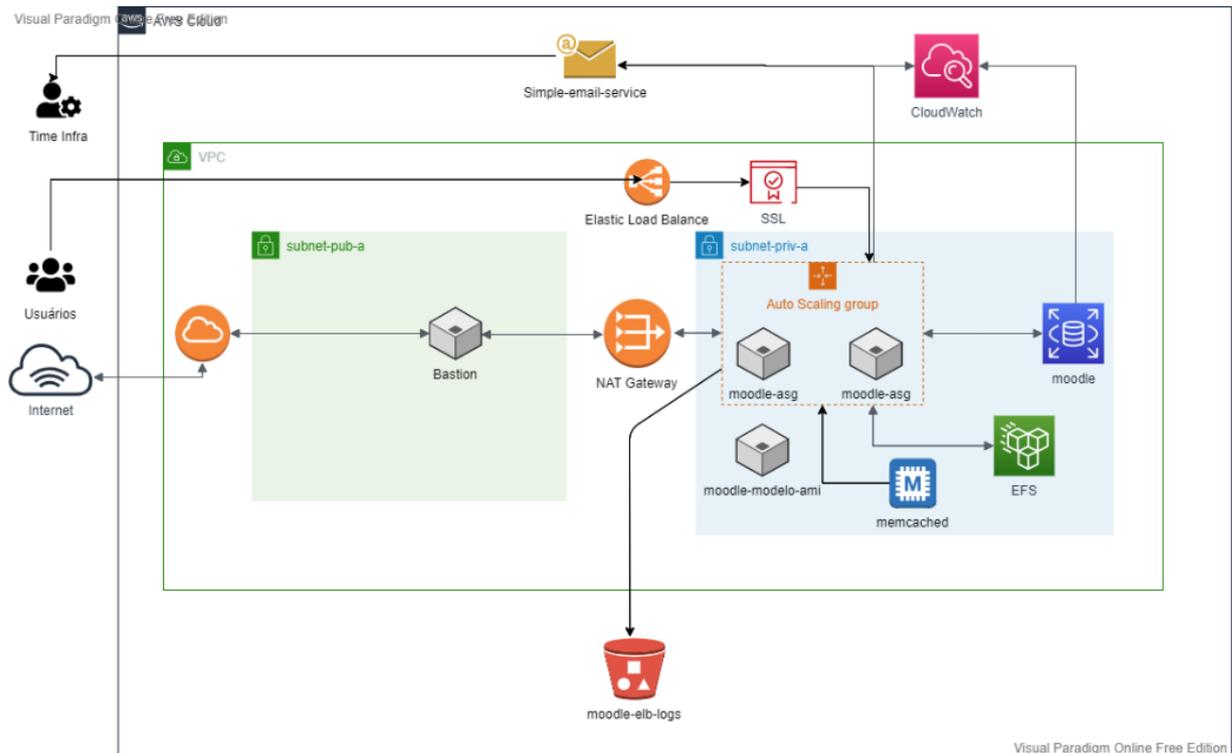
Segundo a AWS, “As instâncias reservadas fornecem economias significativas nos custos do Amazon EC2 em comparação com a definição de preço de instância sob demanda” (AWSReserva, S.I.).

Com a reserva de instância os custos podem reduzir em até 70%. Seu uso é de no mínimo um ano podendo se estender em três caso necessário (AWSReserva, S.I.).

3 Arquitetura na AWS

Nesta seção será apresentada uma arquitetura fácil de criar que poderá servir de modelo para configuração da infraestrutura na AWS e que já atenderia a demanda de altos acesso que as instituições estão tendo com o ensino *online*. Para exemplificar a figura 2 utilizaremos a aplicação *moodle* que é uma aplicação *Open Source* que muitas instituições de ensino utilizam.

Figura 2 Arquitetura sugerida



Fonte: O autor

3.1 Cloudformation

O ambiente da arquitetura apresentado vai precisar de uma VPC com duas *subnets*, uma pública onde todos os serviços que forem colocados nela poderão ser acessados por hosts de fora da AWS e uma *subnet* privada onde os serviços que forem adicionados nessa sub rede só poderá ser acessados por hosts que estiverem na rede interna da VPC criada.

Na arquitetura há também um *NAT Gateway* de Internet para que os servidores que estiverem na *subnet* privada possam ter conexão com a internet. Após sua criação é necessário configurar o *Gateway* adicionando as *subnets* privadas criadas, assim todos os servidores criados nessas sub redes conseguirão se conectar à internet.

3.2 Instâncias EC2

Conforme a arquitetura apresentada deverá estar rodando 4 instâncias no serviço EC2, duas que serão configuradas a partir do grupo de *auto-scaling* com o nome *moodle-asg* onde a aplicação *moodle* rodará, seu tamanho vai depender das necessidades da aplicação que será hospedada nela podendo ser alterado para mais ou para menos se necessário a qualquer momento.

Como elas estão com *auto-scaling* habilitado, caso o número de acessos comece a subir mais do que o normal, o ambiente realizará o escalonamento horizontal, ou seja, mais instâncias iguais as que estão rodando serão criadas conforme for configurado e somente o necessário. Quando a quantidade de acessos diminuir e voltar ao normal o ambiente irá desfazer o escalonamento e o ambiente irá voltar para como estava antes sem a necessidade

da atuação humana. Como sugestão, o ambiente pode ser configurado para ter no mínimo duas instâncias rodando, no máximo oito e duas como ideal, assim o ambiente sempre terá dois servidores *online* no mínimo fazendo com que o tráfego seja dividido evitando que o servidor se sobrecarregue e o escalonamento acontecerá até que tenha 8 instâncias *online*, caso seja necessário o número máximo poderá ser alterado dependendo das regras de negócio da instituição.

Uma instância com o nome de *bastion* pode permanecer desligada sendo ligada apenas quando for necessário acessar as instâncias *moodle-asg*. Por ser uma instância utilizada apenas para fazer um túnel para conexão e nenhuma aplicação estará rodando nela, o tamanho pode ser o menor possível para economizar, e caso a conta na AWS seja nova, ela possui um período de 12 meses de serviços gratuitos e o tamanho t2.micro está incluso no *free tier* fornecido pela AWS sendo o suficiente para essa instância. Na data desse artigo as especificações da instância t2.micro são: 1 vCPU, 6 Créditos de CPU por hora e 1 GB de memória.

Uma última instância que poderá ficar parada, pois será usada apenas para configurações adicionais para a aplicação, testes e atualizações do código fonte. É a partir dessa instância que o time de infraestrutura irá criar um modelo para o grupo de *auto scaling* o nome sugerido é “moodle-modelo-ami”, seu tamanho deverá ser o mínimo exigido pela aplicação podendo ser aumentada ou diminuída de acordo com a necessidade.

Após a criação da instância modelo, o time deverá criar uma AMI dela que é uma imagem do servidor que servirá como modelo para as instâncias auto escaláveis, quando o ambiente precisar se escalonar ele seguirá essa imagem para criar as instâncias.

3.3 Elastic File System - EFS

A arquitetura apresentada sugere a criação de um EFS, sigla para *Elastic File Sysyte*, que servirá para armazenarmos mídias utilizadas pelo ambiente como fotos, vídeos e documentos que serão acessados pelos usuários da aplicação. Esse serviço é necessário, pois como haverá dois servidores e a conexão dos usuários é distribuído entre eles, os arquivos de mídia devem estar iguais para que a experiência do usuário não seja afetada de forma negativa, sendo assim a melhor opção é usar o EFS como um disco compartilhado. Caso contrário, se houver alguma incongruência entre os arquivos pode ser que ao acessar o site algumas mídias não estarem disponíveis. A segunda vantagem de se utilizar o EFS é fazer com que não haja a necessidade de criar um volume EBS muito grande para a instância economizando com o armazenamento.

3.4 ElastiCache

A arquitetura sugere a utilização do serviço *ElastiCache* para a criação de um node de *Memcached* que fornecerá um *endpoint* que deve ser adicionado no arquivo de configuração do php, o *php.ini*, no servidor. Esse serviço armazenará os caches da aplicação melhorando o tempo de resposta do servidor e fazendo a aplicação ficar mais rápida para o usuário, melhorando assim a sua experiência com a plataforma.

3.5 Grupos de segurança

Não está apresentado na arquitetura, mas todas as comunicações entre os usuários e a aplicação deverão estar presentes nos grupos de segurança que servirá como regras de *firewall* permitindo apenas as comunicações necessárias. Na AWS por padrão todo o acesso é bloqueado inicialmente e para permitir a comunicação dos serviços apresentados na arquitetura os grupos de segurança deverão ser criados permitindo a comunicação entre eles.

No caso do projeto apresentado como exemplo a sugestão é fazer a criação de no mínimo quatro grupos, com eles o projeto atenderá o fator de segurança, pois os acessos só serão permitidos para as pessoas autorizadas.

Um grupo de segurança deve liberar o acesso à porta SSH, 22 por padrão, para a faixa de IP utilizada pelo time de infraestrutura da instituição esse grupo deverá ser atribuído apenas à instância *bastion* assim o acesso a ela na porta 22 só poderá ser realizado pelo time de tecnologia da instituição tornando-o mais controlado e seguro.

Um segundo grupo de segurança que libere o acesso a porta utilizada pelo banco de dados que estará hospedado no serviço RDS para a faixa de IPs privado da VPC, assim todas as instâncias que forem criadas na VPC conseguirão se conectar na base de dados. Esse grupo deverá ser atribuído à instância RDS *moodle* tornando o acesso ao banco mais seguro e controlado.

Um terceiro grupo que libere o tráfego na porta 80 e 443 para o coringa 0.0.0.0/0 e que deverá ser atribuído ao *elastic load balance* que for criado, pois assim o acesso de todos os hosts às instâncias *moodle-asg* será permitido. Essa regra é necessária para que os discentes e docentes das instituições possam acessar a aplicação de qualquer lugar que tenha acesso à internet.

Por último um quarto grupo que servirá para permitir acesso as portas 2049, usada pelo EFS, e 11211, usada pelo Memcached, para a rede interna da AWS. Esse grupo é necessário para que os servidores possam se comunicar com os serviços EFS e *Elasticached*.

3.6 Elastic Load Balance

De acordo com a arquitetura, deverá ser criado um *Elastic Load Balance* que servirá como ponte de acesso dos usuários à aplicação via internet, pois os servidores nos quais o site estará hospedada não poderão ser acessados por uma rede de fora da AWS, como mencionado no tópico da VPC, e para que o acesso dos usuários seja criptografado, deverá ser criado um certificado SSL gerado pelo serviço *Amazon Certificate Manager* que garantirá que a aplicação tenha um certificado SSL válido e que os dados dos usuários transportados pela aplicação estejam criptografados, garantido assim a privacidade dos usuários.

Após a criação do ELB, é necessário configurá-lo habilitando o armazenamento dos logs de acesso ao ELB que serão armazenados no *Bucket* que deverá ser criado utilizando o serviço S3 conforme mostrado pela arquitetura. Esses logs poderão ser utilizados pelo time de tecnologia da instituição para analisar um possível problema com a aplicação ou identificar um ou mais IPs mal-intencionados podendo realizar o bloqueio deles se necessário.

3.7 Amazon Certificate Management

Como mencionado anteriormente, deverá ser criado um certificado SSL utilizando o serviço da AWS *Amazon Certificate Management* que deve ser associado ao domínio DNS da aplicação. Como mencionado anteriormente esse certificado garante que todos os dados que forem transportados na aplicação estejam criptografados, evitando assim um ataque do tipo *man-in-the-middle* (Homem no meio).

Nesse tipo de ataque, o hacker consegue ter acesso a tudo que estiver sendo transportado na rede e caso não tenha o certificado SSL, ele terá acesso a informação na íntegra. Já com o certificado essa informação é criptografada, então mesmo que o hacker tenha acesso às informações ele não será capaz de entender o que está sendo transportado, assim a privacidade dos usuários será garantida.

3.8 Relational Database Service – RDS

A arquitetura sugere que seja criado um banco de dados relacional utilizando o serviço RDS assim como as instâncias EC2, o tamanho do RDS poderá ser variado de acordo com a necessidade da aplicação. A criação do RDS é importante para separarmos a base de dados da aplicação, assim o controle sobre ela ficará mais prático e independente.

Como boa prática, o volume dessa instância RDS deverá estar criptografado e com o auto escalonamento do volume habilitado, assim caso o volume comece a atingir o máximo de armazenamento, o escalonamento dele é feito de forma automática garantindo assim que ele não fique fora do ar por falta de espaço em disco.

A maior vantagem para as instituições de ensino em utilizar o RDS é a janela de backup, pois nela pode ser programadas execuções de backups da base de dados diariamente no horário que a instituição achar melhor, assim a academia economizará, pois não será necessário ter um servidor específico de backup e poderá utilizar esses backups para *disaster recovery* caso necessário.

É necessário especificar também uma janela de manutenção que será executada uma vez por semana no dia e horário que a instituição achar melhor, essa janela executará todas as alterações que forem programadas pelo time de infraestrutura como: a alteração do tamanho da instância, alteração de alguma regra e alteração da senha do super usuário por exemplo. Assim não é necessário parar a aplicação em horários de uso para realizar uma alteração e designar uma pessoa para realizar essas alterações, basta programar as alterações e elas serão executadas na janela de manutenção.

A *engine* utilizada no banco ficará a cargo do time de desenvolvimento da instituição, atualmente a AWS tem suporte para as seguintes *engines*:

- *Amazon Aurora*
- MySQL
- MariaDB
- PostgreSQL
- Oracle

- SQL Server

3.9 Armazenamento S3

Como mencionado anteriormente, na arquitetura apresentada é mostrado um *bucket* S3 com o nome de *moodle-elb-logs* que ficará responsável pelo armazenamento dos arquivos de logs de acesso do *Elastic Load Balance* (ELB).

Existem várias opções de armazenamento oferecidas pela AWS são elas: *Standard*, *Intelligent-Tiering*, *Standard-IA*, *One Zone-IA*, *Glacier*, *Glacier Deep Archive* e *Outposts*, sendo o *Standard* a opção padrão da AWS que para a arquitetura apresentada é o suficiente.

Como os arquivos que estarão no *bucket* não serão acessados frequentemente, mas precisam estar acessíveis em pouco tempo quando necessário, a melhor opção de armazenamento é a *Standard-Infrequent Access*, porque o custo de armazenamento e de recuperação serão reduzidos e os objetos estarão disponíveis instantaneamente. A criação de mais *buckets* ficará a cargo da instituição.

3.10 Cloudwatch

Para que o time de infraestrutura possa acompanhar o que está acontecendo com os servidores será utilizado o serviço de monitoramento *CloudWatch* com integração com o SES, serviço de envio de e-mails da AWS.

A seguir vamos fornecer algumas métricas que poderão ser utilizadas e os valores de limites assim quando elas atingirem esse limite a AWS deverá enviar um e-mail alertando o time de infraestrutura sobre um possível problema.

3.10.1 Métricas sugeridas para EC2

As métricas de EC2 sugeridas são:

- *CPU Utilization*
- *CPU Credit Balance* (somente para instâncias da família t)
- *Free Storage Space*
- *Memory Utilization*

3.10.1.1 Cloudwatch Agent

As métricas de *Memory Utilization* e *Free Storage Space* de EC2 não estão presentes por padrão no *cloudwatch*. Para que a coleta dessas métricas seja possível é necessário configurar o *cloudwatch agent* nos servidores. A figura 3 mostra um exemplo do arquivo JSON que deverá ser configurado no servidor na instalação do *cloudwatch agent*.

Figura 3 Arquivo JSON cloudwatch agent

```
{ } arquivoDeConfCWAjson > ...
1
2 {
3   "agent": {
4     "metrics_collection_interval": 60,
5     "run_as_user": "root"
6   },
7   "logs": {
8     "logs_collected": {
9       "files": {
10        "collect_list": [
11          {
12            "file_path": "/var/log/saf-error.log",
13            "log_group_name": "saf-error-log",
14            "log_stream_name": "{instance_id}"
15          },
16          {
17            "file_path": "/var/log/saf-output.log",
18            "log_group_name": "saf-output-log",
19            "log_stream_name": "{instance_id}"
20          },
21          {
22            "file_path": "/var/log/syslog",
23            "log_group_name": "saf-syslog",
24            "log_stream_name": "{instance_id}"
25          }
26        ]
27      }
28    }
29  },
30  "metrics": {
31    "append_dimensions": {
32      "AutoScalingGroupName": "${aws:AutoScalingGroupName}",
33      "ImageId": "${aws:ImageId}",
34      "InstanceId": "${aws:InstanceId}",
35      "InstanceType": "${aws:InstanceType}"
36    },
37    "metrics_collected": {
38      "collectd": {
39        "metrics_aggregation_interval": 60
40      },
41      "disk": {
42        "measurement": [
43          "used_percent"
44        ],
45        "metrics_collection_interval": 60,
46        "resources": [
47          "/"
48        ]
49      },
50      "mem": {
51        "measurement": [
52          "mem_used_percent"
53        ],
54        "metrics_collection_interval": 60
55      }
56    },
57    "aggregation_dimensions": [ ["AutoScalingGroupName"], ["InstanceId", "InstanceType"], [] ]
58  }
59 }
```

Fonte: O autor

Podemos ver no arquivo anterior que as métricas que serão monitoradas pelo agente são duas “used_percent” que está medindo a porcentagem de disco usado no caminho “/” que é o caminho root do servidor e “mem_used_percent” que está medindo a porcentagem de memória que está sendo utilizada. Após a criação desse arquivo, podemos seguir com a criação dos alarmes de “Disk Used Percent” e “Memory Utilization” citados anteriormente.

Como o arquivo de configuração mostra a partir da linha 9, o *CloudWatch* será utilizado para monitoramento dos seguintes logs da aplicação:

- Logs de erros do servidor web Apache error.log
- Logs de acesso do Apache access.log
- Logs do servidor syslog

3.10.1.2 Padrões de alarme sugeridos para EC2

- CPU utilization estiver maior ou igual a 85% por 5 minutos
- Credit Balance estiver menor ou igual a 30% da quantidade máxima por 5 minutos
- Disk Used Percent estiver maior ou igual a 85% por 5 minutos
- Memory Utilization estiver maior ou igual a 90% por 5 minutos

Caso algum desses itens acontecer, a AWS enviará um e-mail para o time de tecnologia da instituição de ensino, assim eles ficarão cientes do que está acontecendo com o ambiente podendo programar alguma alteração caso necessário.

3.10.2 Métricas sugeridas para RDS

As métricas de RDS sugeridas são:

- *CPU Utilization*
- *Burst Balance*
- *Database Connections*
- *CPU Credit Balance* (Somente para instâncias da família t)
- *Free Storage Space*

3.10.3 Padrões de alarme sugeridos para o RDS

- CPU utilization estiver maior ou igual a 75% por 5 minutos
- Burst Balance estiver menor ou igual a 5% por 15 minutos
- Database Connection estiver maior ou igual a 70% do valor máximo permitido por 5 min
- Credit Balance estiver menor ou igual a 30% da quantidade máxima por 5 minutos
- Free Storage Space estiver menor ou igual a 5GB por 5 minutos

Assim como nos alarmes de EC2, caso algum desses itens do RDS aconteça a AWS enviará um e-mail para o time de tecnologia da instituição, assim eles ficarão cientes do que está acontecendo com o banco de dados e poderão realizar alguma alteração case seja necessário.

4 Custos

Vamos apresentar aqui dois possíveis custos para a arquitetura apresentada na seção 3, o custo *On-demand*, que é o padrão cobrado em todos os serviços na AWS. O custo *On-demand* é o valor da arquitetura sem nenhuma economia. Já o outro custo que será apresentado é utilizando reserva de instância, que é um plano oferecido pela própria Amazon que um cliente reserva uma quantidade de servidores por um ou três anos de uso. O plano oferecido pela AWS pode gerar uma economia de até 70% do valor *On-demand*.

Os custos que serão apresentados a seguir foram feitos através da calculadora disponibilizada pela própria AWS que nos permite ter uma estimativa do valor mensal de uma arquitetura na AWS.

4.1 Custo *On-demand*

A figura 4 mostra o custo aproximado do ambiente sem nenhuma opção de economia fornecida pela AWS.

Figura 4 Custo On-demand

Services		Estimate of your Monthly Bill (\$ 212.03)	
Estimate of Your Monthly Bill			
<input checked="" type="checkbox"/> Show First Month's Bill (include all one-time fees, if any)			
Below you will see an estimate of your monthly bill. Expand each line item to see cost breakout of each service. To save this bill and input values, click on 'Save and Share' button. To remove the service from the estimate, jump back to the service and clear the specific service's form.			
Export to CSV		Save and Share	
<input type="checkbox"/> Amazon EC2 Service (US East (N. Virginia))			\$ 141.65
Compute:	\$ 127.85		
EBS Volumes:	\$ 13.80		
EBS IOPS:	\$ 0.00		
EBS Throughput:	\$ 0.00		
<input type="checkbox"/> Amazon RDS Service (US East (N. Virginia))			\$ 52.08
DB instances:	\$ 49.78		
Storage:	\$ 2.30		
<input type="checkbox"/> Amazon Elastic Load Balancing (US East (N. Virginia))			\$ 18.30
Classic LBs:	\$ 18.30		
<input type="checkbox"/> Amazon CloudWatch Service (US East (N. Virginia))			\$ 0.00
Standard Alarms:	\$ 0.00		
<input type="checkbox"/> AWS Support (Basic)			\$ 0.00
Support for all AWS services:	\$ 0.00		
Total Monthly Payment:			\$ 212.03

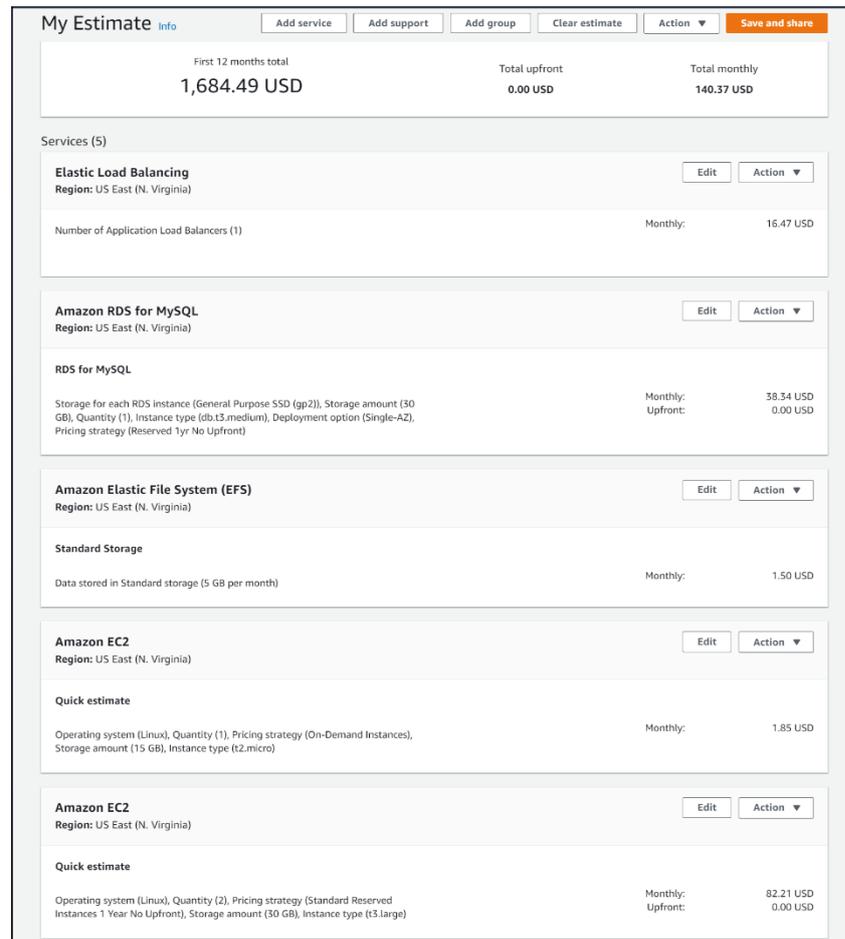
Fonte: O autor

Conforme a calculadora nos mostra, o preço estimado *On-demand* será de U\$ 212.03 considerando o ambiente descrito na arquitetura da seção 3. Na cotação do dólar no dia 11/03/2021 o valor em reais seria: R\$ 1.176,77

4.2 Custo com reserva de instância

A figura 5 mostra o custo mensal aproximado do ambiente utilizando reserva de instância de 1 ano para os servidores EC2 e RDS.

Figura 5 Custo reserva de instância



Fonte: O autor

Conforme a calculadora nos mostra o preço estimado utilizando reserva de instâncias de 1 ano para EC2 e RDS será de U\$ 140.37 considerando o ambiente descrito na seção 3 totalizando U\$ 1,684.49 em 1 ano. Esse valor pode ser pago mensalmente ou o total de uma só vez, assim não será gerado novos custos de EC2 e RDS durante o período da reserva. Com a reserva de instância há uma economia de U\$ 71.66 em relação ao valor *On-demand* o que equivale a 33% do valor sem economia. Na cotação do dólar no dia 11/03/2021 o valor em reais seria: R\$ 779,05

5 Conclusão

Como a computação em nuvem está cada vez mais comum no dia a dia das pessoas e com o aumento do ensino *online* devido a pandemia é de extrema importância que as instituições educacionais comecem a utilizar esse tipo de arquitetura, pois mesmo quando o ensino voltar 100% presencial os benefícios de se utilizar a nuvem entre eles, alta disponibilidade, segurança e custo-benefício continuarão a valer a pena para as entidades educacionais.

O intuito desse artigo foi apresentar a viabilidade de se utilizar a computação em nuvem no setor educacional mostrando que ele pode aproveitar e

muito dos benefícios desse tipo de arquitetura não só durante a pandemia, mas mesmo depois que ela se acabar.

Foi apresentado também um exemplo de arquitetura em nuvem, que pode ser usada por qualquer instituição educacional que deseje implementar a computação em nuvem, bastando apenas de um ou mais profissionais capacitados para aplicar essa arquitetura podendo se aprofundar ainda mais como colocar todos os sistemas utilizados pela universidade na nuvem melhorando assim a segurança e auditoria da instituição. Todas as instituições educacionais deveriam se preocupar em ter uma infraestrutura segura, com alta disponibilidade e que tenha o melhor custo-benefício para essas instituições e com certeza as que aderirem a essa arquitetura não irão se arrepender.

Referências

ROBU, M. *Cloud computing based information systems – present and future. USV Annals of Economics & Public Administration*, v. 12, n. 2, p. 94-100, 2012

YOO, C. S. *Cloud Computing: architectural and policy implications. Review of Industrial Organization*, v. 38, n. 4, p. 405-421, 2011.

AMAZON WEB SERVICE. [S.I.]. O que é computação em nuvem. Disponível em: https://aws.amazon.com/pt/what-is-cloud-computing/?nc1=f_cc. Acesso em: 04 Abril. 2021, 14:36

AMAZON WEB SERVICE. [S.I.]. O que é Amazon VPC?. Disponível em: https://docs.aws.amazon.com/pt_br/vpc/latest/userguide/what-is-amazon-vpc.html. Acesso em: 08 Maio. 2021, 10:32

AMAZON WEB SERVICE. [S.I.]. Grupo de segurança para a VPC. Disponível em: https://docs.aws.amazon.com/pt_br/vpc/latest/userguide/VPC_SecurityGroups.html. Acesso em: 08 Maio. 2021, 10:45.

AMAZON WEB SERVICE. [S.I.]. Documentação do Amazon Elastic Compute Cloud. Disponível em: https://docs.aws.amazon.com/pt_br/ec2/?id=docs_gateway . Acesso em: 04 Abril. 2021, 14:40

AMAZON WEB SERVICE. [S.I.]. O que é o Amazon EC2?. Disponível em: https://docs.aws.amazon.com/pt_br/AWSEC2/latest/UserGuide/concepts.html. Acesso em: 04 Abril. 2021, 15:03

AMAZON WEB SERVICE. [S.I.]. Controlling Network Access to EC2 Instance Using a Bastion Server. Disponível em: <https://aws.amazon.com/blogs/security/controlling-network-access-to-ec2-instances-using-a-bastion-server/#:~:text=A%20bastion%20is%20a%20special,to%20your%20other%20EC2%20instances.&text=To%20define%20the%20source%20IPs,the%20instance%27s%20security%20group%20rules>. Acesso em: 08 Maio. 2021, 11:00.

AMAZON WEB SERVICE. [S.I.]. Elastic Load Balancing. Disponível em: <https://aws.amazon.com/pt/elasticloadbalancing/?whats-new-cards-elb.sort-by=item.additionalFields.postDateTime&whats-new-cards-elb.sort-order=desc>. Acesso em: 08 Maio. 2021, 11:26.

AMAZON WEB SERVICE. [S.I.]. Auto Scaling Group do. Disponível em: https://docs.aws.amazon.com/pt_br/autoscaling/ec2/userguide/AutoScalingGroup.html. Acesso em: 08 Maio. 2021, 11:52.

AMAZON WEB SERVICE. [S.I.]. Endereço IP elásticos. Disponível em: https://docs.aws.amazon.com/pt_br/AWSEC2/latest/UserGuide/elastic-ip-addresses-eip.html. Acesso em: 08 Maio. 2021, 14:29.

AMAZON WEB SERVICE. [S.I.]. O que é o Amazon Relational Database Service (Amazon RDS). Disponível em: https://docs.aws.amazon.com/pt_br/AmazonRDS/latest/UserGuide/Welcome.html. Acesso em: 06 Abril. 2021, 18:14

AMAZON WEB SERVICE. [S.I.]. Documentação do AWS Identity and Access Management. Disponível em: https://docs.aws.amazon.com/pt_br/iam/?id=docs_gateway. Acesso em: 11 Abril. 2021, 18:30

AMAZON WEB SERVICE. [S.I.]. O que é o IAM?. Disponível em: https://docs.aws.amazon.com/pt_br/IAM/latest/UserGuide/introduction.html. Acesso em: 11 Abril. 2021, 18:53

AMAZON WEB SERVICE. [S.I.]. Documentação do Amazon Simple Storage Service. Disponível em: https://docs.aws.amazon.com/pt_br/s3/?id=docs_gateway. Acesso em: 11 Abril. 2021, 09:53

AMAZON WEB SERVICE. [S.I.]. O que é o Amazon S3?. Disponível em: https://docs.aws.amazon.com/pt_br/AmazonS3/latest/userguide/Welcome.html. Acesso em: 11 Abril. 2021, 10:15

AMAZON WEB SERVICE. [S.I.]. Documentação do Amazon CloudWatch. Disponível em: https://docs.aws.amazon.com/pt_br/cloudwatch/?id=docs_gateway. Acesso em: 11 Abril. 2021, 11:08

AMAZON WEB SERVICE. [S.I.]. Coletar métricas e logs de instâncias do Amazon EC2 e de servidores no local com o agente do CloudWatch. Disponível em: https://docs.aws.amazon.com/pt_br/AmazonCloudWatch/latest/monitoring/Install-CloudWatch-Agent.html. Acesso em: 24 Maio. 2021, 20:58

AMAZON WEB SERVICE. [S.I.]. O que é o Amazon CloudWatch?. Disponível em: https://docs.aws.amazon.com/pt_br/AmazonCloudWatch/latest/monitoring/WhatIsCloudWatch.html. Acesso em: 14 Abril. 2021, 21:50

AMAZON WEB SERVICE. [S.I.]. O que é o AWS Certificate Manager?. Disponível em: https://docs.aws.amazon.com/pt_br/acm/latest/userguide/acm-overview.html. Acesso em: 08 Maio. 2021, 14:47

NETO, Manoel Veras de Souza. Computação em Nuvem. Rio de Janeiro. 2015

AMAZON WEB SERVICE. [S.I.]. Documentação do Elastic File System. Disponível em: https://docs.aws.amazon.com/pt_br/efs/?id=docs_gateway. Acesso em: 29 Junho. 2021 19:39

AMAZON WEB SERVICE. [S.I.]. O que é AWS Comand Line Interface. Disponível em: https://docs.aws.amazon.com/pt_br/cli/latest/userguide/cli-chap-welcome.html. Acesso em: 29 Junho. 2021 19:50

AMAZON WEB SERVICE. [S.I.]. Documentação do Amazon ElastiCache. Disponível em: https://docs.aws.amazon.com/pt_br/elasticache/?id=docs_gateway. Acesso em: 29 Junho. 2021 20:00

AMAZON WEB SERVICE. [S.I.]. Instâncias Reservadas. Disponível em: https://docs.aws.amazon.com/pt_br/AWSEC2/latest/UserGuide/ec2-reserved-instances.html. Acesso em: 29 Junho. 2021 20:15