

Insper

Engenharia de Computação

Entrega do relatório de Cloud

Pedro Cliquet do Amaral

Aplicação AWS via CloudFormation

São Paulo

2024

Documentação técnica

Introdução

O projeto consiste em criar um script que automatiza o processo de criar uma nuvem na AWS. Dessa forma, o desenvolvimento consiste em utilizar ferramentas da AWS para que a aplicação esteja de acordo com as seguintes especificações:

- Tenha Balanceamento de Carga
- Gerenciamento via CloudFormation
- Criação, edição e destruição completa da infraestrutura

Para isso, é importante destacar os recursos existentes na AWS que serão utilizados no desenvolvimento do projeto e o custo atribuído para aplicabilidade.

Recursos utilizados da AWS:

1. IAM:

O *IAM* é utilizado para criar e gerenciar usuários, configurando permissões de usabilidade dos recursos contidos na AWS. Após criar seu usuário é importante saber quais acessos foram atribuídos caso queira criar novas aplicações via AWS. Ele garante que segurança caso mais pessoas queiram acessar recursos internos com intuito de apenas verificação do sistema.

Pensando como uma empresa, criar uma conta que possui a função de administrador, tenha permissão de criar até 5 usuários e possa fazer 1000 solicitações por mês no API Check o valor a ser pago é em torno de 3 dólares por mês.

No caso do projeto, o usuário criado terá acesso aos recursos abaixo:

- Administrativo
- DynamoDb
- EC2
- S3
- CloudFormation
- EC2 Instance Connect
- ELB
- IAM FullAccess

Com as permissões podemos alterar e verificar os componentes criados via script, ou seja, monitorar o funcionamento da aplicação, já que o usuário possui permissões para isso.

2. EC2:

O Recurso *EC2* é escolhido devido a sua capacidade de criar e gerenciar instâncias com artifícios de segurança criptográfica. Nele, o usuário é capaz de criar '*n*' instâncias de servidor, selecionar o tipo de sistema operacional, a capacidade de armazenamento e processamento e conexão com a rede.

Esse recurso é capaz de acessar os servidores e instalar aplicativos ou serviços para utilizá-las na arquitetura da aplicação.

No projeto foram criados os seguintes recursos via EC2:

- VPC
- Subnet
- Internet Gateway
- RouteTable
- Route
- SubnetRouteTableAssociation
- Security Group

No EC2, para contabilizar os gastos considera-se as seguintes especificações:

- Instâncias Compartilhadas
- OS Linux
- Uso constante
- 1 Instância

Instância do EC2:

- Tipo: t2.medium
- vCPUs: Any
- Memória: Any
- Desempenho de rede: Any

Utilizando o plano de Compute Savings Plans, o custo mensal será de 16.86 dólares.

3. ELB - ALB

O ELB possui diversos recursos para balanceamento de carga entre aplicativos e servidores. No caso, o recurso do ELB escolhido foi o ALB, que está relacionado com o funcionamento da camada 7, direcionando o tráfego para aplicativos baseados em HTTP/HTTPS. Neste recurso cria-se os seguintes componentes:

- Load Balancer(LB)
- Listener
- Target Group

O LB é o recurso citado anteriormente, chamado no projeto de *Application Load Balancer*, que servirá de referência, em conjunto com o *Target Group*, para o *Listener*. O *Listener* representa a porta em que a nossa aplicação estará disponível para escutar requisições via internet. O *Target Group* tem a tarefa de checar os protocolos de acesso e comunicar com o *Listener* via porta configurada. Por fim, ele referencia o VPC que será explicado a seguir.

Para a precificação do ELB, considera-se as seguintes opções:

- Application Load Balancer: Ativo
- Número de Aplicações: 1
- 1 Gb hora de processamento na instância
- Número médio de novas conexões via ALB: 2/s
- Tempo de duração de conexão: 10 minutos
- Número médio de solicitações por segundo: 1000
- Número médio de avaliação de regras por solicitação: 10

Com essas especificações, o custo mensal do EC2 será de 22,27 dólares.

4. VPC

O VPC (Virtual Private Cloud) é um recurso da AWS que cria uma rede virtual em nuvem da AWS. Esse recurso é proveniente com o EC2 e servirá de referenciamento dos seguintes recursos criados pelo EC2:

- Subnet
- Gateway
- Router
- InstanceSecurityGroup
- ALBSecurityGroup
- TargetGroup
- DynamoDB EndPoint

O VPC permitirá que os recursos de *Gateway*, *Tabelas de roteamento*, *sub-redes* e *Security Groups* possam atuar. Basicamente, o S3, EC2 e o DynamoDB precisam de uma VPC para funcionar corretamente.

Dessa forma, a custo será contabilizado pelas seguintes informações:

- VPN Connection
- Gateway Load Balancer
- Network Address Translation (NAT) Gateway
- Public IPv4 Address

VPN Connection:

- Número de conexões do Site-to-Site VPN: 1
- Duração média de cada conexão: 24 horas/dia

Configuração do Client VPN

- Número de Subredes: 1
- Número de conexões(ou usuários) do Cliente VPN ativas: 1/dia
- Duração média de cada conexão: 1 horas/dia
- Dias úteis por mês: 22

Gateway LB:

- Número de zonas de disponibilidade nas quais está implementado: 1

Transit Gateway:

- Número de anexos: 1
- Dados processados: 1GB/mês
-

O cálculo do gasto total, para a VPC, é a soma de todas as configurações. Neste caso deram 196,44 dólares por mês.

5. DynamoDB

O *DynamoDB* é um banco de dados que irá guardar os inputs da aplicação, efetuado pelos usuários que acessarem o site via HTTP. É possível alterar as configurações dele para que sua locação de dados seja dinâmica, ou seja, caso necessite ele aumenta a capacidade de armazenamento automaticamente.

Os gastos com o banco de dados é estimado pelos seguintes recursos:

- DynamoDB on-demand capacity
- DynamoDB Data import from Amazon S3
- DynamoDB Data export from Amazon S3

Armazenamento de dados:

- Capacidade: 10GB
- Tamanho médio dos itens: 2 KB

DynamoDB Data Export:

- 10 GB de exportação
- 10 GB de exportação incremental

O custo total foi de 4.50 dólares.

6. CloudWatch

O recurso do CW é utilizado para prevenir a propagação de erros das máquinas para as aplicações. Assim, ele garante segurança para que o usuário não se depare com um possível problema que ocorreu na aplicação.

Para a configuração total do CW tem-se:

- APIs
- Logs
- Consulta de logs
- Painéis de Alarmes
- Canários
- *CW Logs*
- Insights DynamoDB
- Insights Lambda
- Evento do Rum
- Monitor de Internet
- Sinais de aplicações

Atribuindo o menor valor possível para todos os recursos, o que não garante qualidade no serviço, tem-se o valor de 21.05 dólares.

7. AWS CloudFormation

O CloudFormation é um recurso da AWS que auxilia a criação de aplicações em conjunto com monitoramento, permitindo subir na AWS por meio de arquivos. Essa ferramenta permite que scripts sejam criados e, de forma rápida, sejam implementados de acordo com as especificações do arquivo, que pode ser no formato json ou yaml.

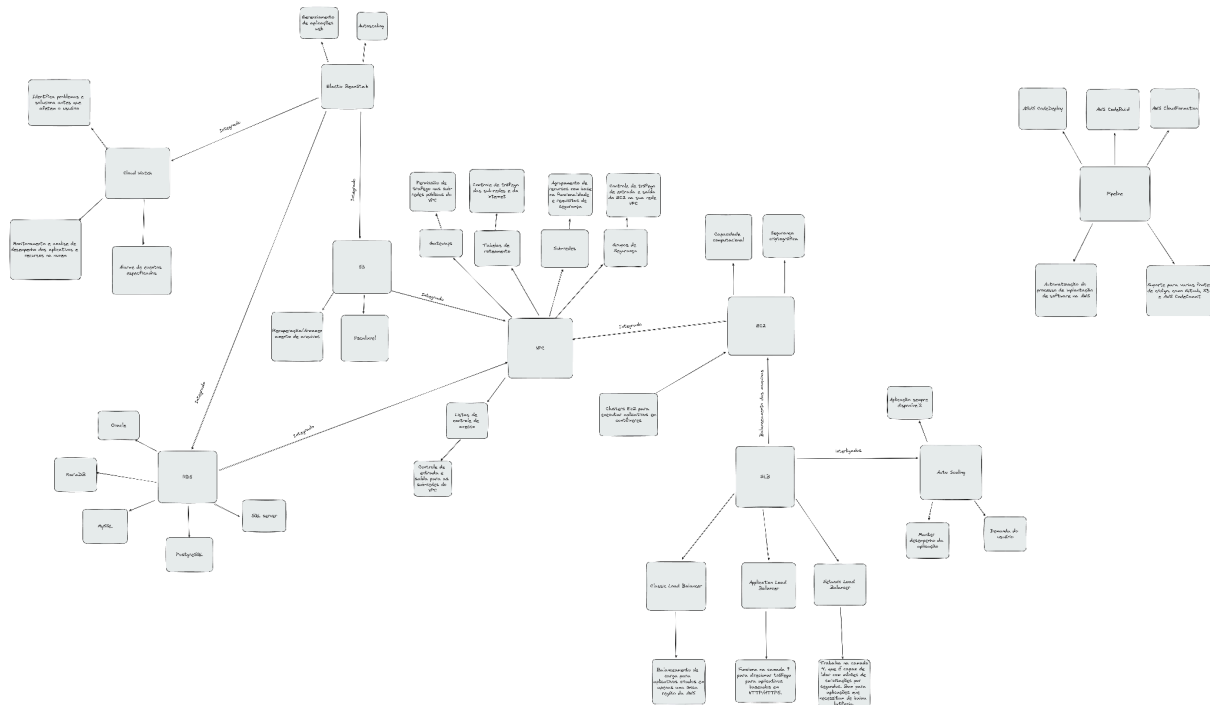
As configurações de do *CF*, para estimativa de preços:

- Número de extensões de terceiros gerenciadas: 100
- Número total de extensões: 10 /dia
- Duração média por operação: 30 segundos

Com isso, obtém-se um valor de 26.48 dólares.

Custo total: 290.6 dólares por mês.

Diagrama geral dos recursos AWS:

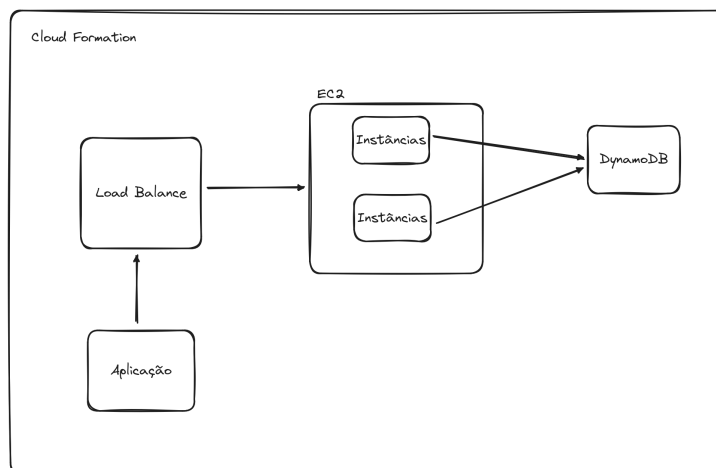


Para melhor visualização:

https://excalidraw.com/#json=PvRP6HIX3c63v0gFnBula.7VgyTaTle_TKa4vCnAqvGA

Esse diagrama descreve as interações e funções de cada componente utilizado no projeto, para auxiliar o entendimento, efetue a leitura dos *Recursos utilizados da AWS*, no início do documento, enquanto analisa o diagrama. O diagrama também auxilia o leitor a entender o script que gera a infraestrutura. No script, cada componente possui uma ligação com outro, identificado pelo prefixo: “!Ref”. Isso significa que o componente depende de outro para que a aplicação funcione corretamente.

Segue abaixo o diagrama simplificado da aplicação:



Utilização do Script

Para subir a aplicação via script pelo *Cloud Formation* deve primeiro possuir um arquivo *.yaml* ou *.json*, no repositório a seguir, foi utilizado o formato *.yaml*:

- https://github.com/pcliquet/AWS_Cloud.git

Em seguida, utiliza-se o recurso *IAM* para criar um usuário, que possa usufruir dos requisitos necessários para rodar o script baixado no link, e que tenha permissões para gerenciar via *CloudFormation*. As permissões de uso foram citadas no *Recursos utilizados da AWS*, porém segue os mesmos abaixo:

- Administrativo
- DynamoDb
- EC2
- S3
- CloudFormation
- EC2 Instance Connect
- ELB
- IAM FullAccess

Em seguida, o usuário deverá instalar o *AWS Cli* em sua máquina. No caso desse projeto, o sistema operacional é o ubuntu, portanto os códigos para instalar via prompt seguem abaixo:

1. Instalar dependências

```
sudo apt-get update  
sudo apt-get install -y unzip - caso não tenha o unzip em sua máquina
```

2. Baixar os arquivos do AWS CLI

```
curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
```

3. Descompactar o arquivo

```
unzip awscliv2.zip
```

4. Instalação

```
sudo ./aws/install
```

5. Verificação

```
aws --version
```

Com o *AWS Cli* em sua máquina, deve-se configurar as credenciais dentro dele utilizando o comando:

```
aws configure
```

Nele deve-se fornecer as seguintes informações:

- AWS Access Key ID: Chave de acesso encontrada no recurso do IAM dentro do usuário que quer subir o script
- AWS Secret Access Key: Essa chave deve ser gerada no perfil do usuário e não pode ser compartilhada. Devido a essas condições, a AWS disponibiliza essa senha apenas 1 única vez, portanto save-a bem!
- Default Region name: Região onde quer que sua aplicação esteja alocada
- Default output format: Formato de saída padrão(“json” ou “.yaml”)

Após a configuração das credenciais, é necessário criar uma pilha para que os componentes presentes no script sejam executados. Essa pilha requer os seguintes items:

- stack-name: o nome da stack que deseja criar
- template-url: url que contém o arquivo para subir os componentes.
- parameters: parâmetros necessários pelo seu template.

No caso do script do projeto, acesse a pasta onde o repositório está clonado e rode o comando abaixo:

```
aws cloudformation create-stack --stack-name amaral-stack --template-body  
file://main.yaml --parameters ParameterKey=SSHLocation,ParameterValue=0.0.0.0/0  
--capabilities CAPABILITY_IAM
```

Para saber se a aplicação funcionou, acesse a AWS e entre nas opções do *EC2*. Nesse dashboard, procure pelo componente *Load Balancer*, e copie o *ID* gerado. Com esse *ID*, abra uma nova aba do navegador e cole, ele irá te direcionar para a página da aplicação. Caso sua aplicação utilize um banco de dados, faça uma requisição do tipo “POST”, pela sua aplicação, acesse o dashboard do *CloudFormation* e procure pelo banco de dados que usou, no caso do projeto foi o *DynamoDB*. Dentro do *DynamoDB*, procure por *Tables* e *items*. Se o banco de dados estiver alinhado com a aplicação, nessa aba aparecerá o input efetuado dentro da sua aplicação.

Conclusão

O serviço da *AWS*, permite que faça aplicações de modo rápido e seguro, o domínio completo dos recursos permite que projetos possam ser implementados de forma rápida. Entretanto, a falta de conhecimento pode gerar gastos excessivos e inviáveis para tipos diferentes de empresas. Com base na estimativa feita nesse projeto, é possível concluir que os gastos feitos de forma intuitiva, e não planejada, geram valores altos devido a qualidade dos recursos oferecidos. Portanto, o estudo aprofundado de cada componente pode aliviar os gastos totais para empresas de pequeno porte, ou empresários autônomos.