# Projeto Appoio Digital - AGES 2020/2

Este documento possui possui informações **úteis sobre AWS**, em uma versão genérica do documento que entregamos aos stakeholders do projeto Appoio em 2020/2.

Autores: Alessandro Medeiros, Bianca Camargo, João Severo, Rafael Ritter e Rafael Ruwer.

# 1. Observações gerais

A estrutura básica para o funcionamento da aplicação se resume em: API (Application Programming Interface) hospedada na AWS com os serviços que são descritos ao longo deste documento e versão de produção do aplicativo utilizando o endereço da EC2.

Sempre que a instância EC2 for parada, o endereço de conexão do aplicativo para a instância, é modificado automaticamente e então uma nova versão do aplicativo precisa ser gerada, utilizando o endereço novo. A instância pode ficar parada para evitar consumo dos recursos que podem gerar cobranças.

# 2. Amazon Web Services (AWS)

A Amazon Web Services (AWS) é a plataforma de nuvem mais adotada e mais abrangente do mundo, oferecendo mais de 175 serviços completos de datacenters em todo o mundo. Milhões de clientes, incluindo as startups de crescimento mais rápido, grandes empresas e os maiores órgãos governamentais, estão usando a AWS para reduzirem seus custos, ficarem mais ágeis e inovarem mais rapidamente.

Mais informações: https://aws.amazon.com/pt/

# 3. Serviços Utilizados

No desenvolvimento do projeto Appoio foram utilizados três serviços da AWS: EC2, RDS e SES.

### **3.1.** Elastic Compute Cloud (EC2)

O Elastic Compute Cloud (EC2) é o serviço que provê capacidade computacional redimensionável na nuvem da Amazon.

Mais informações: https://aws.amazon.com/pt/ec2/

### 3.1.1. Tipos de Instância

O Amazon EC2 oferece uma ampla seleção de tipos de instâncias otimizadas para atender a diferentes casos de uso. Os tipos de instâncias consistem em várias combinações de CPU, memória, armazenamento e capacidade de rede e oferecem flexibilidade de escolha da composição adequada de recursos para as suas aplicações. Cada tipo de instância inclui um ou mais tamanhos de instância, permitindo a escalabilidade de seus recursos de acordo com os requisitos da carga de trabalho a ser executada.

As instâncias T2 são um tipo de instância de uso geral de baixo custo. As instâncias T2 são uma das opções de instância do Amazon EC2 de menor custo, sendo ideais para uma variedade de aplicações de uso geral. As instâncias t2.micro são elegíveis para o Free Tier (nível gratuito) da AWS.

Mais informações: https://aws.amazon.com/pt/ec2/instance-types/

## 3.2. Relational Database Service (RDS) para MySQL

O MySQL é o banco de dados relacional de código aberto mais popular do mundo e o Amazon RDS facilita a configuração, a operação e a escalabilidade de implantações de MySQL na nuvem. Com o Amazon RDS, você pode implantar em minutos servidores MySQL escaláveis com capacidade de hardware econômica e redimensionável.

O Amazon RDS para MySQL gerencia as tarefas de administração de banco de dados demoradas, incluindo backups, aplicação de patches, monitoramento, escalabilidade e replicação, permitindo que você se concentre no desenvolvimento de aplicações.

Mais informações:

- <u>https://aws.amazon.com/pt/rds/</u>
- <u>https://aws.amazon.com/pt/rds/mysql/</u>

### **3.3.** Simple Email Service (SES)

O Simple Email Service (SES) é o serviço de envio de e-mails na nuvem da Amazon.

Mais informações: https://aws.amazon.com/pt/ses/

# 4. Free Tier

O Free Tier (nível gratuito) da AWS depende de uma série de fatores, como o serviço que está sendo utilizado. Entre os tipos ofertas gratuitas disponíveis, existem as seguintes:

- **Sempre gratuito:** Estas ofertas de nível gratuito não expiram e estão disponíveis para todos os clientes da AWS.
- **12 meses gratuitos:** Aproveite estas ofertas por 12 meses, a contar da data de seu cadastramento inicial na AWS.

Mais informações: https://aws.amazon.com/pt/free/

## 4.1. Elastic Compute Cloud (EC2)

#### Tipo de oferta: 12 meses gratuitos.

O nível gratuito da AWS inclui 750 horas de instâncias t2.micro todo mês durante um ano.

Mais informações: https://aws.amazon.com/pt/ec2/pricing/

## 4.2. Relational Database Service (RDS) para MySQL

Tipo de oferta: 12 meses gratuitos.

O Amazon RDS pode ser testado gratuitamente. Pague somente pelo que usar. Não há taxa mínima.

O nível gratuito da AWS inclui:

- 750 horas por mês de uso do banco de dados db.t2.micro;
- 20 GB de armazenamento de banco de dados de uso geral (SSD); e
- 20 GB de armazenamento para backups de banco de dados.

Mais informações: https://aws.amazon.com/pt/rds/mysql/pricing/

### 4.3. Simple Email Service (SES)

Tipo de oferta: Sempre gratuito.

O Amazon Simple Email Service (SES) é um serviço pago conforme o uso com base no volume de e-mails enviados. Não há assinaturas, negociações de contrato nem cobranças mínimas.

Quando você chama o Amazon SES de uma aplicação hospedada no Amazon EC2, você pode enviar 62.000 mensagens por mês sem custo adicional. Esse benefício do nível de uso gratuito não expira. Depois disso, a cobrança é de 0,10 USD para cada 1.000 e-mails recebidos.

Mais informações: https://aws.amazon.com/pt/ses/pricing/

# 5. Deploy do Backend no ambiente AWS

Para fazer o deploy da nossa aplicação fizemos uso da plataforma Amazon pela facilidade de uso e quantidade de recursos oferecidos pela empresa. Com isso precisamos possuir uma conta na Amazon Web Services para poder subir a nossa aplicação. Para criar uma conta basta acessar o <u>endereço deste</u> <u>link</u> e criar gratuitamente a sua conta AWS.

# 5.1. Criando instância EC2

Com a conta criada vamos para a instanciação da nossa máquina EC2, para isso siga os passos a seguir:

- 1. Acesse a sua conta AWS
- 2. Clique no botão "Services" no canto superior esquerdo da tela
- 3. Na aba de "Compute" selecione a opção "EC2"
- 4. Procure na janela que abriu o botão escrito "Launch Instance"
- Agora nós vamos configurar a nossa instância, utilize a navegação lateral para clicar na opção "AWS Marketplace"
- 6. Com a opção selecionada, procure por "ECS", irão aparecer algumas opções, selecione a "ECS Optimized Amazon Linux 2"
- 7. Na próxima seção iremos escolher o tipo de instância, caso se deseje continuar no nível gratuito, escolha a opção t2.micro e clique em "Review and Launch"
- 8. Clique em "Launch"
- 9. Agora vamos criar (caso já não possua) um par de chave-valor para acessar a nossa instância
- 10. Para clicar no botão "Create a new key pair" para criar uma nova e informar no campo "Key pair name" o nome da sua chave-valor.
- 11. Após configurado faça download do arquivo chave-valor, pois será utilizado para acessar a nossa máquina.
- 12. Para completar selecione "Launch Instance".

## 5.2. Configurando portas de acesso

Pronto, você acaba de criar uma instância EC2, vamos configurar a segurança para possibilitar que um cliente externo possa acessar a nossa aplicação. Vamos fazer essa configuração agora:

- 1. Na página inicial das instâncias EC2 selecione a máquina que acabamos de criar para alterar o grupo de segurança
- 2. Ao selecionar, apareceu uma janela embaixo com algumas informações detalhadas da máquina, procure por "Security Groups" e clique no nome ao lado que deve estar em azul
- 3. Na parte superior da página, clique em "Actions" e, em seguida, clique em "Edit Inbound Rules". Na caixa de diálogo que se abrir, clique em "Add Rule"
- 4. Defina o intervalo de portas da regra para 3000 e, em "Source", clique no menu suspenso e selecione "Anywhere". O restante deve ser preenchido automaticamente.

Pronto, definimos regras de acesso à máquina e precisamos configurar a nossa aplicação na máquina, para isso nós vamos precisar acessar a nossa instância.

## 5.3. Subindo o projeto na instância

Siga os passos a seguir de acordo com a sua plataforma

#### Linux

- 1. chmod 400  $\leq$  NAME\_OF\_KEYPAIR\_FILE>
- 2. ssh -i <NAME\_OF\_KEYPAIR\_FILE>ec2-user@<PUBLIC\_DNS>
- 3. git clone <u>http://tools.ages.pucrs.br/appoio/backend.git</u>

#### Windows

- 1. Download da ferramenta putty: Link para download 64-bit Link para download 32-bit
- 2. Download da ferramenta puttygen: Link para download 64-bit Link para download 32-bit
- 3. Utilize a ferramenta puttygen para transformar a nossa chave-valor de .pem para .ppk clicando no botão "Load" e selecione o arquivo .pem
- 4. Após carregar o arquivo clique em "Save private key" e coloque o nome que quiser
- 5. Agora abra a ferramenta putty e informe no campo "Host Name (or IP address)" o endereço da sua instância, deve ser semelhante a "ec2-user@ec2-35-163-140-132.us-west-2.compute.amazonaws.com", para dúvidas de acesso navegue até a página da sua máquina e clique no botão "Actions" e em seguida "Connect", nan aba de "SSH" terá o endereço da máquina
- 6. Com o endereço informado, navegue, no putty, para "Connection" > "SSH" > "Auth", não clique em "GSSAPI", clique em "Auth" e no campo "Private key file for authentication" informe o arquivo gerado pelo nosso puttygen
- 7. Com tudo configurado clique em "Open"
- 8. git clone http://tools.ages.pucrs.br/appoio/backend.git

Pronto, o backend da nossa aplicação já está no nosso servidor da AWS, agora precisamos fazer algumas configurações para contemplar todas as funcionalidades da nossa aplicação (envio de e-mail e configuração das variáveis de ambiente)

# 5.4. Configurando o AWS Simple Email Service (SES)

Esta configuração é necessária para que, quando um novo tutorial seja enviado ou dados sejam exportados, seja possível enviar a informação para o e-mail do administrador, para isso nós precisamos seguir um passo a passo simples:

- 1. Acesse a página inicial da sua conta AWS e procure pelo serviço Simple Email Service
- 2. Ao selecionar o serviço irá aparecer uma nova tela com diversas opções no canto esquerdo, selecione a opção "Email addresses"
- 3. Na nova janela clique no botão "Verify a New Email Address"
- 4. No popup insira o seu e-mail que será utilizado como remetente e clique no botão "Verify This Email Address".
- 5. Você receberá um e-mail no e-mail informado para confirmar a sua identidade, siga conforme solicitado no e-mail
- 6. Realize o passo 1 novamente para acessar o serviço Simple Email Service
- 7. Ao selecionar o serviço irá aparecer uma nova tela com diversas opções no canto esquerdo, selecione a opção "Sending Statistcs"
- 8. Caso não tenha sido feita esta configuração ainda, deve haver um botão escrito "Edit your account details", clique nele
- 9. Na janela que abrir, preencha os dados de acordo com a sua necessidade, para esta aplicação foi feita a seguinte configuração:

**Enable production access:** Yes

Mail type: Transactional

Website URL: <endereço da instância>

**Use case description:** Sending emails functionality is used to export data and notify only the owner of the application, no emails are sent to customers.

### Additional contact addresses: Preferred contact language: English

10. Após o preenchimento do formulário aceite os termos e clique no botão "Submit for review". A equipe da Amazon irá analisar o pedido e conceder ou recusar a permissão

Pode ser necessário enviar mais formulários conforme for requisitado pela equipe da Amazon

Agora temos o serviço de e-mails da Amazon configurado, mas ainda precisamos de mais umas configurações para poder utilizá-lo.

## 5.5. Configurando Credenciais de Acesso

Essa credencial é utilizada para que você acesse a sua conta da Amazon e utilize os serviços disponíveis nela, para isso vamos aos seguintes passos:

- 1. Acesse a sua conta AWS
- 2. Clique em cima do seu nome do canto superior direito, irão aparecer algumas opções
- 3. Selecione a opção "My Security Credentials"
- 4. Na nova janela que abriu procure por "Chaves de acesso (ID da chave de acesso e a chave de acesso secreta)", deve estar no meio da tela, clique em cima
- 5. Caso não tenha salvo nenhuma chave de acesso clique no botão "Criar nova chave de acesso"
- 6. Uma nova chave de acesso foi criada automaticamente, faça o download, pois não será possível recuperar o segredo da chave depois

Prontinho, agora temos tudo que precisamos para criar nosso arquivo de variáveis de ambiente

## 5.6. Criando o Arquivo de Variáveis de Ambiente

Este arquivo irá conter todas as variáveis de ambiente que nossa aplicação irá utilizar, nele vamos colocar as seguintes informações:

- SECRET: uma senha aleatória para criptografar as sessões do nosso aplicativo
- ADMIN\_EMAIL: e-mail do administrador (o mesmo que verificamos anteriormente e que receberá os e-mails de aviso e de dados)
- ADMIN\_PASSWORD: senha do e-mail do administrador
- API\_URL: o endereço da nossa instância (é o campo "Public DNS (IPv4)" da nossa instância EC2)
- AWS\_SES\_ACCESS\_KEY\_ID: o campo AWSAccessKeyId da chave de acesso que fizemos o download anteriormente
- AWS\_SES\_SECRET\_ACCESS\_KEY: o campo AWSSecretKey da chave de acesso que fizemos download anteriormente
- AWS\_SES\_REGION: a região em que está localizada a nossa instância (ao acessar o serviço EC2 deve aparecer no canto superior direito uma cidade, no caso desta aplicação é Ohio, se clicar em cima irá aparecer uma lista com uma posição geográfica, a cidade e uma região em laranja, no nosso caso é "Leste dos EUA (Ohio) us-east-2") o que precisamos é somente o

us-east-2 (no nosso caso, pode ser que a sua instância esteja em outro local, veja o nome que está em laranja no seu para garantir a região correta)

- DATABASE\_NAME: o nome do nosso banco de dados que criamos
- DATABASE\_USERNAME: o usuário que cadastramos no nosso banco de dados
- DATABASE\_PASSWORD: a senha que cadastramos no nosso banco de dados
- DATABASE HOST: o endereço da instância que possui o banco de dados
- DATABASE\_PORT: a porta em que está executando o nosso banco de dados
- CLOUDINARY\_URL: a URL do nosso serviço no Cloudinary

Todas essas informações devem ser inseridas no arquivo env.list no formato

<CHAVE>=<VALOR>

Sendo que chave irá ser substituído sem os símbolos de < e > pelos campos mencionados acima, e o valor, sem os símbolos de < e >, pelos seus respectivos valores. Um exemplo de arquivo de configuração com valores falsos pode ser encontrado na pasta de exemplos.

Se todas as etapas até agora foram realizadas, meus parabéns, você conseguiu configurar com sucesso a sua instância da AWS.

# 5.7. Comandos do Docker para Criar e Verificar o Projeto

Alguns comandos úteis para executar o Docker dentro da sua instância, ressalto que todos os comandos devem ser executados dentro da pasta do projeto:

Para criar a imagem do Docker que será executada utilize o comando:

• docker build . -t backend

Para executar a imagem criada utilize o comando:

• docker run -d -p 3000:3000 --env-file env.list backend

Note que a porta de acesso do docker mapeia para a porta 3000, ou seja, a porta definida na nossa aplicação deve ser igual a porta informada no docker, neste caso 3000

Para matar a instância do Docker criada, use o comando:

• docker kill < tag da instância>

#### Sem os símbolos de $\langle e \rangle$

Para verificar as instâncias executando no Docker utilize o comando:

• docker stats

Para verificar o que foi exibido no terminal da instância do Docker utilize o comando:

• docker logs <tag da instância>

Sem os símbolos de < e >

# 6. Desligando a instância

Para desligar a instância EC2 basta seguir os seguintes passos:

- 1. Acesse o seu console da AWS https://console.aws.amazon.com/console/home
- 2. Abra a janela de Services e localize o serviço EC2, clique sobre EC2
- 3. Na nova janela que abriu, utilize a barra lateral de navegação para ir até Instances
- 4. No painel de Instances você poderá ver todas as instâncias criadas na sua conta
- 5. Neste painel, selecione as instâncias que deseja para a execução (no canto esquerdo de cada instância tem uma checkbox)
- 6. Com todas instâncias selecionadas, clique no botão Actions localizado mais ao centro da tela
- 7. No menu que abrir terá a opção Instances State, com o mouse sobre a opção irá mostrar todos os estados em que é possível colocar as instâncias selecionadas, ou seja, se todas estiverem executando é possível paralisar todas as instâncias ou terminá-las (destruir a instância). O mesmo ocorre caso as instâncias estejam todas paralisadas, é possível inicializá-las ou terminá-las.