

Especificação do Trabalho Prático 1 - Uso da AWS
Desenvolvimento de Software para Nuvem
Professores: Dr. Fernando A. M. Trinta / Dr. Paulo A. L. Rego

CONTEXTUALIZAÇÃO

Nas últimas duas décadas observamos o nascimento da Computação em Nuvens e o seu amadurecimento como um paradigma computacional. Hoje a Computação em Nuvens é vista por muitos como uma solução para muitos dos problemas enfrentados por empresas, órgãos governamentais e academia.

De forma bastante resumida, a principal característica da Computação em Nuvem está relacionada ao conceito de computação como um serviço, ou seja, trata-se da ideia de que infraestrutura, hardware e software possam ser “alugados” e utilizados pelos usuários como um serviço, de forma semelhante ao que ocorre com serviços tradicionais do nosso cotidiano como, por exemplo, os serviços de disponibilização de energia elétrica ou de água, o serviço computacional pode ser adaptado conforme a necessidade do usuário (quanto maior a demanda, mais recursos são alocados para o serviço prestado ao usuário). A tarifação também é proporcional a demanda do usuário, ou seja, quanto mais o usuário consome, maior será o valor a ser pago pelo serviço prestado.

Relacionados a esta forma de serviço, temos os papéis dos provedores e dos consumidores de serviço. Os provedores de serviço são aqueles que possuem os recursos computacionais e os disponibilizam a terceiros para uso. Comumente, provedores de serviço são grandes empresas do ramo da Tecnologia da Informação (TI) como Google, IBM e Microsoft. Já os consumidores de serviço são aqueles que utilizam os recursos disponibilizados pelos provedores. Neste caso, há uma maior diversidade nos tipos de consumidores, que podem ser desde pessoas físicas até grandes empresas do ramo de TI ou não.

Hoje, um dos principais provedores de Computação em Nuvem do mercado é a Amazon com a AWS (*Amazon Web Services*¹). A AWS é uma plataforma que oferece um conjunto significativo de serviços através da Internet. Estes serviços vão desde a disponibilização de infraestrutura para alocação e utilização por terceiros, até a oferta de recursos de software para uso de outras aplicações (como, por exemplo, o AWS Lambda e Amazon SageMaker). Para este trabalho, se faz necessária uma melhor contextualização de quatro dos diversos serviços disponibilizados no AWS: Amazon EC2², Amazon S3³, Amazon RDS⁴ e DynamoDB⁵.

Amazon EC2 é o principal e o serviço de IaaS (Infraestrutura como Serviço) da AWS. Neste serviço, o usuário solicita a criação de instâncias virtuais. Em geral, o usuário especifica

¹ <https://aws.amazon.com/pt/>

² <https://aws.amazon.com/pt/ec2/getting-started/>

³ <https://aws.amazon.com/pt/s3/getting-started/?nc=sn&loc=5&dn=1>

⁴ <https://aws.amazon.com/pt/rds/resources/>

⁵ <https://aws.amazon.com/pt/dynamodb/getting-started/>

configurações de hardware e software da instância (CPU, Memória RAM, Disco, Sistema Operacional, Firewall, ...) e o Amazon EC2 cria a instância com base nessas configurações. Uma vez criada, ela oferece meios de acesso a esta instância virtual recém criada como, por exemplo, via SSH ou console próprio da AWS.

Amazon S3 é um dos serviços de *storage* como um serviço da AWS. Neste serviço, o usuário cria instâncias de armazenamento (denominadas *Buckets*). Dentro dos *Buckets*, é possível criar uma hierarquia de diretórios e pastas visando uma maior organização dos arquivos. Por fim, Amazon RDS e Dynamo DB são serviços de banco de dados como serviço da AWS. RDS e Dynamo DB são serviços, respectivamente, de Banco de Dados Relacional e Não-Relacional da AWS. Ambos os serviços atuam permitindo a configuração e o gerenciamento simplificado dos bancos de dados e oferecem suporte aos principais bancos de dados do mercado.

OBJETIVO

O objetivo deste trabalho é avaliar, de forma prática, o conhecimento adquirido pelos alunos referente ao desenvolvimento de programas em Nuvem. Para isso, foi escolhido a Nuvem da EC2 como ambiente para hospedar a aplicação e o uso de alguns dos principais serviços disponibilizados pela AWS para compor as funcionalidades da aplicação (S3, RDS e DynamoDB).

DESCRIÇÃO GERAL DO SISTEMA

A primeira parte da atividade consiste em implementar **um sistema simplificado de gerenciamento de patrimônio**. Em geral, o sistema deverá ser composto por dois grandes módulos: um de gerenciamento de usuários e um de gerenciamento de bens. O primeiro será o responsável por cadastrar novos usuários, bem como atualizá-los ou removê-los. Já o segundo estará relacionado à função principal do sistema: implementar um mecanismo que permita que os usuários possam gerenciar bens de uma instituição/empresa.

Através do sistema de gerenciamento de usuários, cada usuário pode se cadastrar no sistema, atualizar seus próprios dados cadastrais ou sair do sistema. Por questões de segurança, não se deve permitir que usuários acessem ou manipulem qualquer tipo de dado de outros usuários. O sistema também deve dispor de usuário super-admin (root) que pode realizar todas as operações para todos os usuários, ou seja, um usuário logado como root poderá cadastrar, atualizar e remover usuários.

O segundo módulo, o de gerenciamento de bens, terá as funcionalidades descritas a seguir. Primeiramente, um usuário quando regularmente autenticado no sistema poderá cadastrar um bem, enviando arquivo que comprove a existência do bem (nota fiscal de compra ou imagem do bem em uma sala da instituição/empresa, nos formatos [.pdf](#), [.jpeg](#) ou [.png](#)), nome, número de patrimônio e localização do bem. Usuários podem inserir, consultar por nome e localização,

alterar os dados e remover bens. Qualquer falha deve ser informada ao usuário em uma tela personalizada indicando o motivo da falha (Ex: erro no upload do arquivo, etc).

Para cada bem cadastrado, o sistema deve gerar um código único que o identifica e que será utilizado para validação futura. O sistema deve permitir que bens sejam validados por usuários externos. Para isso, o sistema deve fornecer uma página pública em que qualquer pessoa (não há necessidade de ser um usuário cadastrado do sistema) possa consultar um bem a partir do seu código de identificação. Ao fazer este acesso, o usuário externo validará ou não o bem. Esta informação deve ser registrada pelo sistema. Podem haver mais de uma validação de um mesmo bem por diferentes usuários.

Quando o usuário logado visualizar a lista de bens, ele deve ter as informações do número de vezes que o bem foi validado/consultado através do seu código único. Além disso, ele pode visualizar por meio de um log: data e hora das validações com sucesso e sem sucesso, bem como quais eram as motivações de cada uma destas validações.

PARTE 1

Para implementar este sistema na AWS, os alunos devem utilizar as seguintes ferramentas:

1. **Amazon EC2:** Nesta ferramenta, devem ficar hospedados os servidores da aplicação. Assim, os alunos devem instanciar uma ou mais máquinas virtuais e executar nelas a aplicação. A aplicação deverá interagir com as outras ferramentas da AWS durante seu ciclo de vida.
2. **Amazon S3:** Nesta ferramenta, devem ser persistidos os arquivos submetidos pelos usuários (pdf, jpg ou png). Por questão organizacional, cada usuário deve possuir uma pasta pessoal na ferramenta para armazenar seus próprios arquivos.
3. **Amazon RDS:** Nesta ferramenta, devem ser persistidos os dados dos usuários e de seus arquivos. Os atributos dos usuários são: (CódigoUsuário, Nome Completo, identificação do usuário, senha e e-mail). O Código é o campo chave e deve ser produzido pela própria aplicação. Os demais dados devem ser informados pelo usuário via formulário. Já outra tabela deve conter a lista de arquivos de um usuário (CódigoArquivo, CódigoUsuário, NomeArquivo). Com esta tabela, o nome do arquivo deve apontar para um bucket no S3 onde o arquivo é armazenado.
4. **Dynamo DB:** Nesta ferramenta, devem ser persistidas as informações necessárias para logar as validações de documento. Assim, cada entrada deverá ter o identificador único como chave e o código do arquivo validado, se a validação foi positiva ou não e a motivação para validação).

O sistema deve permitir que se façam as seguintes operações:

1. CRUD de usuário (Criar e excluir usuários, bem como requisitar e atualizar seus dados);

2. CRUD de arquivos de usuários. Observação, quando um arquivo for removido ou alterado, suas validações devem também ser removidas
3. Realizar validações de arquivos de acordo com código do arquivo existente e arquivo fornecido, além da motivação para a validação.;
4. Listar na página principal do usuário, seus arquivos indicando o número de validações positivas e negativas. Caso o usuário deseje, ele pode ter o relatório completo das tentativas de validação para cada arquivo;

PARTE 2

Parte 2 – A partir da aplicação desenvolvida na Parte 1, utilizar os Serviços de **LoadBalancing** e **Auto Scaling** para tornar a aplicação elástica.

Configuração e Regras para Elasticidade:

- a) A elasticidade será conseguida pela estratégia horizontal. Inicialmente deve haver apenas uma instância da aplicação, executando em uma instância do tipo *micro*.
- b) Um balanceador de carga deve ser colocado à frente da(s) instância(s) da aplicação, de modo a distribuir a carga de trabalho entre as instâncias.
- c) Caso a média de uso de CPU dessa instância exceda 70% por mais de um minuto, deverá ser instanciada uma nova instância para dividir a carga de trabalho da aplicação. O número de instâncias pode crescer até no máximo 3 (três).
- d) Caso a média de uso da CPU do conjunto de instâncias fique abaixo de 10% por mais de um minuto, uma instância deve ser finalizada.

Getting Started:

Auto Scalling Group: <https://aws.amazon.com/pt/ec2/autoscaling/getting-started/>

Vídeos Tutoriais sobre Auto Scaling Group:

<https://www.youtube.com/watch?v=7SfVZqOVcCI>

<https://www.youtube.com/watch?v=vNic2xziwIY>

Algumas ponderações:

- 1) A aplicação deve possuir uma interface Web, Desktop ou Mobile para uso de suas funcionalidades;
- 2) A comunicação com os serviços da AWS deve ser feita utilizando API REST;
- 3) Caso a aplicação seja implementada como uma aplicação Web, a mesma deve ser executada em uma máquina virtual da Amazon EC2.