

## Projeto - APS - Individual

### Um sistema ORM multi-cloud com Load Balancer e Autoscalling.

- Implementar comunicação via REST API.
- Autenticação de usuário stateless.
- Sistema de log das atividades.
- Possuir um aplicação cliente.
- Possuir um script de implantação.
- Precisa apresentar individualmente o funcionamento do projeto durante a semana da avaliação final.
- Entrega: Até a data da Aval. Final (26/Nov/2021).

## ToDo

Pessoal segue um ToDo para a realização do Projeto. Antes algumas considerações:

1. Escolher uma plataforma. Alguns sugestões Python+Boto3 (maioria), Nodejs+AWS SDK, Ansible, Containers (Kubernetes ou Docker).
2. Comece um passo de cada vez. Ao invés de escrever um script enorme e tentar caçar bugs de uma vez, comece com passos simples e incremente, garatinfo que todas as etapas estão funcionando.
3. Organize as tarefas, monte uma lista do que fazer.

# Inspêr

## ToDo:

1. Crie uma instância em Ohio usando a plataforma. Se você vai usar Boto3 por exemplo, utilizar a documentação oficial: [boto3.readthedocs.io](https://boto3.readthedocs.io). Dica: Você precisa destravar a role no IAM para lançar algo em Ohio.
2. Uma vez criada a instância, voce precisa instalar um banco de dados. Você pode usar o que foi feito no H2 e instalar o Postgres. Dica: Usar o argumento UserData na criação da instância para rodar comandos na inicialização. Dica: o UserData não roda no \$HOME. Dica: Você precisa rodar os comandos sem prompt, por exemplo da senha. Dica: você pode escolher outro banco de dados como o MySQL, Maria ou Mongo.
3. Após instalar o banco, garanta que ele esteja funcionando e que esteja acessível. Use um client (psql por exemplo para o Postgres) em outro local para verificar. Dica: Você vai precisar liberar o SecurityGroup.
4. Crie uma segunda instância em outra região (North Virginia). Instale o ORM nessa máquina (H2 novamente), apontando para o BD criado no item 2. Dica: você precisa de 2 clients no boto3 para isso, um para cada região.
5. Testar o ORM para ver se tudo está funcionando direito. Dica: Security Group de novo.
6. Crie uma AMI da segunda instância, depois destrua-a.
7. Criar um Load Balancer + Autoscaling group usando a AMI do item 6. Teste o LB.
8. Fazer o client para consumir os endpoints do terminal. Dica: usar Python com a biblioteca requests.
9. Nesse ponto você atingiu o C. Revise seu script para conseguir rodar 2 vezes sem intervenção manual (basicamente capturar os IPs

automaticamente e destruir os itens antes de criá-los) para atingir o C+. Dica: Use TAGS nos itens para filtrar mais facilmente.

10. Modifique o script para instalar um VPN server na instância do BD e VPN clients na AMI para criar a conexão VPN. Isso qualifica B+. Dica: OpenVPN.

11. Retire o ORM do Autoscaling e coloque em uma nova instância em Ohio. Instale o VPN Server nessa instância agora.

12. Crie uma instância em North Virginia que possui apenas um webserver que recebe um request HTTP e repassa para o ORM. Instale o VPN Client nessa máquina. Isso será chamado de Gateway.

13. Crie páginas HTML para interagir com o Gateway. Crie um webserver que servem essas páginas e coloque no autoscaling. Isso atingirá o A+.

Ainda, caso implementem o sistema de logs (guardar as ações dos usuários) e um sistema de autenticação stateless, cada item soma 1/2 conceito na nota.

Falta pouco para terminar o semestre, foquem em terminar o H3 em sala, o H4 é composto apenas de perguntas. Não se esqueçam do H0 também. Ainda, não falem nas aulas da manhã, a adesão a essas aulas tem sido baixíssima (<50%). Lembrando ainda: H0 e projeto tem caráter de APS, portanto devem ser majoritariamente feitos nas horas alocadas fora do horário de aula (esse semestre temos pelo menos 28h extras alocadas para essa disciplina). Ainda dá tempo para terminar, mas será preciso se organizar e usar as horas extras alocadas agora.

**Rodolfo Avelino**

Professor Insper