

**Nosso cliente atualmente tem um sistema ERP instalado em um servidor on premise e devido a problemas na infra do prédio o sistema ficou indisponível muitas vezes, impactando na operação do negócio e prejudicando o acesso via VPN dos funcionários em home office.**

**Nosso projeto tem como objetivo a Migração deste sistema ERP para a Nuvem AWS com backup do banco de dados na nuvem Google Cloud.**

## **AWS**

### **IAM**

**Na nossa conta Root AWS:**

- **Habilitamos o autenticação Multi Fator**
- **Aumentamos a política de senha**
- **Criamos um grupo de usuário ADM**
- **Criamos 5 usuários ADM, um para cada integrante do grupo**

**Conforme a necessidade liberamos as funções para o usuário, mantendo controle de acesso aos serviços da AWS.**

### **VPC**

**Criamos 1 VPC e 3 subnets (uma subnet para cada Zona de Disponibilidade).**

**2 grupos de segurança , uma para as instâncias de aplicação e outra para as instâncias de banco de dados do RDS, para separar os tipos de acesso a portas.**

**Criamos 1 route table e adicionamos à ela a VPC e Subnets, e por fim associamos um Internet Gateway para que a nossa aplicação tivesse saída para a Internet.**

### **EC2**

**A partir da primeira Instância de Aplicação EC2 que criamos, tiramos uma imagem (AMI) para criar mais 2 iguais. Totalizando 3, uma em cada zona de disponibilidade.**

**Habilitamos o Load Balance Application para balancear a carga de acesso externo a aplicação entre as 3 instâncias.**

## **S3**

**Criamos um bucket no S3 para armazenar o backup do banco de dados e fazer o deploy da aplicação nas instâncias EC2.**

## **CodeDeploy**

**Utilizamos o CodeDeploy para executar automaticamente a instalação da aplicação nas instâncias EC2.**

## **RDS**

**Criamos um banco de dados relacional mysql utilizando o serviço de RDS com replicação multi AZ, para manter a alta disponibilidade e tolerância a falhas do banco de dados.**

**Tiramos um snapshot do banco de dados e exportamos para o bucket dentro S3.**

## **Auto scaling EC2**

**Primeiro criamos um Modelo de Execução adicionando a mesma AMI (imagem) da aplicação e o mesmo modelo de instância.**

**Em seguida configuramos o Grupo de Autoscaling utilizando este modelo de execução e adicionamos nele a VPC e Subnets da UAI Cloud, foi habilitado o Load Balance Application e as verificações de integridade a cada 30 seg. Capacidade desejada nas instâncias de app executando é de 3, no mínimo 1 e no máximo 6 executando.**

**A política de escalabilidade foi definida para quando a média do uso da CPU das instâncias atingirem 80%, a partir de 60 seg, uma nova instância igual ser aquecida para execução.**

## **Na plataforma Google Cloud**

### **Cloud Storage**

**Criamos um bucket no cloud storage para armazenar as cópias dos backups do banco de dados da aplicação.**

## **Storage Transfer Service**

**Utilizamos serviço de transferência de dados da GCP para fazer uma cópia do backup do banco de dados do bucket S3 da AWS.**

### **Aplicação**

**Acessando o link <http://uaicloud.96ti.com.br> poderam dar uma olhada na aplicação Dolibarr de código aberto que subimos para testar a aplicação.**