# AD on Cloud with Server less Technology for USA based Medical Product and Equipment Company

**Area of Engagement:**
**AWS Serverless Technologies**
**Geo:**
**USA**

## Engagement Background

Customer wanted to migrate their exiting Dealer Portal application to AWS using Serverless technologies provided by AWS. Dealer portal application allows different class of users to see different information about the product based on their requirement. This restriction is built by providing different menu option for different class of users. User class is defined during the user creation process. The current application was written in PHP and was hosted by external service provider. New application is rewritten using Agular 5 and NodeJS and hosted completely on AWS. The product search result had to be restricted based on the user roles. Only admin user can see all the products and other users sees filtered search result based on their role. Product detail page information display restriction was implemented for different user based on role. The application provides Admin to import, export product information from the front end. Admin can also create users and export user information to a file from front end. The manual process of daily inventory update to the database had to be automated.

## Business Challenges

- Migrate the application to company domain name.
- Automate daily inventory feeds received from SAP to the new application. Current process is manual.
- Filter information displayed about products to users based on role.
- Restrict application functionality based on user roles.
- Categories different class of users based on roles during user creation.

## Solution

- Rewrite the current application written in PHP using Angular 5 and NodeJS 8.
- Use AWS RDS Aurora Postgres for Database.
- Use AWS API Gateway for creating API's.
- AWS Lambda is used to write application logic using NodeJS.
- User authentication is implemented using AWS Cognito user pools and federated identities.
- User role based feature implemented using AWS Cognito, API gateway and Lambda. That is different users access different functionality based on the access.
- Lambda functions determines the roles provided by Cognito authorization tokens.
- Lambda function returns different result based on role.
- RDS is to be encrypted.
- RDS access credentials is encrypted using AWS KMS.
- All the lambda functions and RDS is inside the VPC without public access.
- Rebuild the database tables and migrate data.
- Automate the process of updating inventory details in the database which currently was manual.
- Use S3 trigger to invoke a lambda function to update the Database with inventory information. S3 triggers lambda function whenever inventory information in CSV file formation is uploaded to the S3 bucket.
- AWS CloudFront is used for CDN

## Benefits to Customer

- Build the application using serverless technologies which is creating a lot of interest in the IT world.
- Automate daily inventory feeds received from SAP to the new application. Current process is manual.
- Application build using modern programming languages such as Angular and NodeJS.
- Control the access to the user based on role.

## Measurable outcome

- All the existing functionality retained.
- Automated manual inventory detail update to database.
- New UI created.

## Tools and Technology

AWS API gateway, Cognito, RDS Aurora (Postgres), KMS, S3, Cloudfront