# AWS Implementation Plan

**Phase 1: Initial Setup Verification and Optimization (Week 1)**

1. **Verify Existing Resources and Configurations**
   * **Task:** Take a snapshot of the current MySQL, database and setup of current EC2 instances as well as the integrated website.
   * **Roles:** Cloud Solution Architect
   * **Duration:** 1 day
   * **Deliverables:** The Resource Inventory Report can also be used as one of the results to show the values before the implementation of the intended strategies.
2. **Optimize Database Configuration**
   * **Task:** This should be done in order to make MySQL optimized for provision of its services. Check OS type, the type of storage media/optimal storage using Amazon RDS SSD and connection preference.
   * **Roles:** Database Administrator
   * **Duration:** 2 days
   * **Deliverables:** Trying to fine tune an un-optimised database can sometimes prove to be a frustrating experience, specially to those who may not fully understand the inner workings of such a system.
3. **Review Security Configurations**
   * **Task:** IAM roles needs to be audited to ensure that they have the least level of permission required; as well as Security Groups and Network ACLs for compliance purposes.
   * **Roles:** Security Specialist
   * **Duration:** 2 days
   * **Deliverables:** Security Audit Report

**Phase 2: Scaling and Reliability Enhancement (Week 2)**

1. **Set Up Auto Scaling for EC2 Instances**
   * **Task:** Apply Auto Scaling Groups so that your EC2 instances can adjust their quantity according to the traffic of your website.
   * **Roles:** Cloud Solution Architect
   * **Duration:** 2 days
   * **Deliverables:** This is a documentation on Auto Scaling Configuration.
2. **Deploy Elastic Load Balancer (ELB)**
   * **Task:** One must create an ELB to balance the incoming traffic across the different EC2 instances to gain availability and fault tolerance.
   * **Roles:** Network Architect
   * **Duration:** 2 days
   * **Deliverables:** This ELB Setup tutorial will enable everyone to understand how to go through the process to achieve the end result.
3. **Enable CloudFormation for Infrastructure as Code (IaC)**
   * **Task:** They should employ AWS CloudFormation to create templates that can be used in the setup of infrastructure hence reducing the handling and deployment time of resources.
   * **Roles:** We have a Cloud Solutions Architect for any cloud advice we require for our organization.
   * **Duration:** 3 days
   * **Deliverables:** CloudFormation Templates

**Phase 3: Security and Compliance (Week 3)**

1. **Implement AWS WAF and Shield**
   * **Task:** Set up AWS WAF to shield your web app against common web-layer exploits as well as turn on AWS Shield for DDoS guard.
   * **Roles:** Security Specialist
   * **Duration:** 3 days
   * **Deliverables:** The next in the list is the definition of WAF and Shield Configuration Documentation.
2. **Database Security Hardening**
   * **Task:** Secure the data in the MySQL database by providing encryption of the data which are stored or which are in transit. In AWS it is recommended to make use AWS Key Management Service (KMS) to manage encryption values.
   * **Roles:** Database Administrator, Security Specialist
   * **Duration:** 2 days
   * **Deliverables:** The report which focuses on the security of databases.
3. **Regular Compliance Checks**
   * **Task:** To help detect noncompliance and track activities within your AWS environment set up AWS Config and AWS CloudTrail.
   * **Roles:** Security Specialist
   * **Duration:** 2 days
   * **Deliverables:** In terms of compliance and monitoring its configuration can also be done easily depending on the needs of the users.

**Phase 4: Continuous Integration and Deployment (CI/CD) (Week 4)**

1. **Set Up AWS CodePipeline**
   * **Task:** Use AWS CodePipeline for your website’s continuous integration solution and continuous deployment for the testing and deployment of your application.
   * **Roles:** DevOps Engineer
   * **Duration:** 4 days
   * **Deliverables:** documentation of CI CD Pipeline
2. **Containerize Applications Using ECS**
   * **Task:** If necessary, package your application using Docker and host it to AWS ECS since the platform has a higher level of scalability.
   * **Roles:** DevOps, Applications development team
   * **Duration:** 5 days
   * **Deliverables:** Some of the sources that could be consulted in relation to ECS includes the ECS Deployment Guide and Docker Image Repositories.

**Phase 5: Monitoring and Cost Management (Week 5)**

1. **Implement AWS CloudWatch**
   * **Task:** Set CloudWatch for monitoring basic metrics of your EC2 instances, RDS and general application performance. Establish alerts for key issues, that would contribute to powerful warning signals.
   * **Roles:** A cloud solutions architect.
   * **Duration:** 3 days
   * **Deliverables:** AWS CloudWatch Dashboard and the Alarm Setup
2. **Set Up AWS Cost Explorer and Budgets**
   * **Task:** AWS Cost Explorer and AWS Budgets should be used as tools to track and monitor spending so as not to use up much of AWS service.
   * **Roles:** Specialized Team of Finance and Cloud.
   * **Duration:** 2 days
   * **Deliverables:** Identify to whom budget alerts will be issued and details of a cost management plan.

**Phase 6: Review and Finalization (Week 6)**

1. **Conduct a Final Review of the AWS Environment**
   * **Task:** Review all the configuration settings, all the performance parameters and all the security options to make sure everything is working properly.
   * **Roles:** In All team that was formed when the action was launched have been mentioned.
   * **Duration:** 2 days
   * **Deliverables:** I – Final Review Report
2. **Finalize Documentation**
   * **Task:** Capture all the documentation done in the project in a reference manual to be used in the future apart from the auditing process.
   * **Roles:** The key team roles are a Project Manager and the Cloud Solutions Architect.
   * **Duration:** 2 days
   * **Deliverables:** AWS Documentation for the Complete Environment

**Conclusion**

By consultation with AWS experts, Ozmart Retail Group will be in a position to follow this detailed plan and be assured of a strong, elastic, and highly secure AWS cloud infrastructure necessary for the company’s cloud needs in the future. This entails periodic monitoring of the infrastructure, or performing security audits, and more importantly, optimization which will enable one to keep running the infrastructure at the lowest possible cost while at the same time achieving the highest levels of performance.

========================================================================================================================================================

For purposes of creating a sound cloud environment for Ozmart Retail Group that is reliable, scalable and secure, the outcomes highlighted from the case study in the documents attached previously will be adopted. Thus, the case study describes the vision of cloud migration at Ozmart and a set of measures aimed at improving the company’s IT environment, increasing organizational flexibility, improving data protection, and optimising costs. The following is a breakdown of a detailed plan based on the case study, to attain these objectives using AWS service;

**Case Study Implementation Plan for Ozmart Retail Group**

**1. Objective**

The purpose of this cloud migration project is to effectively migrate Ozmart Retail IT system into cloud environment that is more reliable, agile and secure. The new AWS-based platform will provide Ozmart increased flexibility and handle its growth, strengthened security, and lower overhead while interoperating with Ozmart’s existing legacy applications.

**2. Key Components of the AWS Cloud Environment**

**a. Hybrid Cloud Architecture Design**

1. **Architecture Overview**:
   * **Hybrid Cloud Model:** Extend your company’s on premises computing environment to include integrated links to AWS public cloud computing services to create hybrid cloud. With this model, Ozmart is able to take advantage of the factors of cloud environment such as scalability and flexibility while at the same time retaining important on premise operations.
   * **Components:** The primary ones are the AWS servicemodels such as EC2 for computing capacity, S3 for storage, RDS for managed database, VPC for networking and security isolation.
2. **Implementation Steps**:
   * **Set Up AWS Virtual Private Cloud (VPC):** Implement a VPC with public and private subnets in order to ensure that all internet-facing instances are isolated from backend ones.
   * **Configure Secure Connectivity:** AWS Direct Connect in combination with Virtual Private Gateway makes it possible to create secure Links between physical data centers and AWS cloud.

**b. Compute Resources and Auto Scaling**

1. **Elastic Compute Cloud (EC2)**:
   * **Task:** You should leverage EC2 instances for the elastic computing power. Instance types will be chosen based on load they have to handle in order to prove performance and cost effectiveness for the company.
   * **Auto Scaling:** Use ASGs to generate EC2 instances by scaling up or down in accordance with computed demands to facilitate cost optimization.
2. **Implementation Steps**:
   * **Deploy EC2 Instances:** Configure all of EC2 instances for multiple purposes and usages.
   * **Configure Auto Scaling:** Use Auto Scaling Groups and Policies for automatic scaling of instances according to certain set limits.

**c. Storage and Data Management**

1. **Amazon S3 and RDS**:
   * **Task:** For stored objects use S3 and for relational data stored databases you can use RDS. This ensures security, sturdiness and growth of storage space for data, back-ups and log files.
   * **Amazon Elastic File System (EFS):** Use EFS for general purpose but accessing shared file system for highly available file storage that is capable to handle numerous connection requests from different EC2 instances.
2. **Implementation Steps**:
   * **Set Up S3 Buckets:** Develop S3 buckets for Staging Files, Backups, and Log Files.
   * **Deploy RDS Instances:** Deploy the RDS for managed databases that should also have the features that include backup and patching automation.

**d. Networking and Security**

1. **Network Design**:
   * **Task:** Plan and configure the virtual Private Clouds, subnet and Security groups. Use features such as Network ACL and Security Groups to manage traffic, into as well as out of AWS.
2. **Security Measures**:
   * **Identity and Access Management (IAM):** Introducing IAM best practices that guarantee the least privilege access and the protection of available AWS resources.
   * **AWS WAF and Shield:** Use AWS WAF to help protect web applications and AWS Shield to help protect against DDoS attacks.
3. **Implementation Steps**:
   * **Configure VPC and Subnets:** Created VPC, properly distinguished subnets for every tier of the application.
   * **Set Up IAM Roles and Policies:** Identity and access management: develop IAM roles and policies for access.
   * **Implement WAF and Shield:** AWS Web application firewall and Shield should be set up to protect from web application attacks and DDoS.

**e. Application Migration and Optimization**

1. **Application Migration**:
   * **Task:** Modernize applications by ‘Lifting and shifting’ and replatforming, and refactor to make them suitable for PaaS**.**
2. **Continuous Integration and Continuous Deployment (CI/CD)**:
   * **Task:** Integrate deployment through the AWS CodePipeline and CodeDeploy services to enable automatic updates on applications.
3. **Implementation Steps**:
   * **Plan Application Migration:** Subsequently, outline for each application a well thought-out migration plan.
   * **Deploy CI/CD Pipeline**: AWS code pipeline and Code Deploy should be used in order to automate the process of deployment and updating.

**f. Monitoring, Logging, and Cost Management**

1. **Monitoring and Logging**:
   * **Task:** For the purpose of monitoring key metrics, alarms, and notifications, AWS CloudWatch should be used while AWS CloudTrail can be used to trail activities taking place within the AWS ecosystem.
2. **Cost Management**:
   * **Task:** AWS Cost Explorer and AWS Budgets should be implemented to track costs and thence optimize on usage of resources.
3. **Implementation Steps**:
   * **Configure CloudWatch and CloudTrail:** Define the rules of monitoring and logging for all essential AWS services.
   * **Enable Cost Management Tools:** AWS usage reports, AWS Cost Explorer, and AWS Budgets should be used in tracking and managing the costs.

**3. Phased Implementation Plan**

To ensure a smooth transition and minimal disruption, the migration will be implemented in phases:

1. **Phase 1: Preparation and Planning (Weeks 1-2)**
   * **Tasks:** Business requirements definition, risk identification and evaluation, cloud migration assessment, cloud hybrid architecture evaluation.
   * **Deliverables:** For example, a requirements document, risk assessment report, cloud readiness report and deliverables in the form of network architecture diagrams.
2. **Phase 2: Core Infrastructure Setup (Weeks 3-4)**
   * **Tasks:** VPC, EC2, S3, RDS, and EFS: setup, launch instance, integrations, and storage.
   * **Deliverables:** Things like provision of VPC and subnet, documentation of EC2 and RDS, guides to creating S3 and EFS.
3. **Phase 3: Security and Compliance (Weeks 5-6)**
   * **Tasks:** IAM, security group and network ACL setup and WAF and Shield solutions.
   * **Deliverables:** IAM roles and policies, security configurations of systems and applications, WAF, Shield and other setup guide.
4. **Phase 4: Application Migration and Optimization (Weeks 7-8)**
   * **Tasks:** Application migration; configuration and integration, continuous integration and continuous development pipeline; application optimization.
   * **Deliverables:** Migration plan for applications, specification of CI/CD pipeline, results of studying application’s performance.
5. **Phase 5: Testing and Validation (Weeks 9-10)**
   * **Tasks:** Functional and performance testing, security testing; disaster recovery testing.
   * **Deliverables:** Preparation of test reports, documentation of documents as related to disaster recovery.
6. **Phase 6: Go-Live and Post-Migration Monitoring (Weeks 11-12)**
   * **Tasks:** conversion and creation of templates, post goes live activities , training.
   * **Deliverables:** GO live checklist, checking of reports, training completed check list.

**Conclusion**

Thus, by implementing this specific plan grounded on the case study, Ozmart Retail Group can get an AWS cloud, which is secure, scalable and cost-effective. This is because the cloud infrastructure needs to be checked periodically, must be audited, and must be updated frequently so as to serve the company’s purpose effectively.