Prepared by: Mohammad Fahim Nazari

**Facility Pro Serverless Architecture**

**Facility Pro Current Architecture**

Upon reviewing the current architecture of Facility Pro, the following components have been identified:

1. EC2 Instances: There are two EC2 instances deployed in the architecture.
2. VPC (Virtual Private Cloud): The architecture is set up within a VPC, which provides an isolated network environment in the AWS cloud.
3. Public and Private Subnets: The EC2 instances are deployed in separate subnets within the VPC.
4. DynamoDB: DynamoDB is a NoSQL database service provided by AWS. It is being used in the architecture, presumably to store and retrieve data required by the Facility Pro application.
5. Load Balancer: A load balancer is included in the architecture to distribute incoming traffic across multiple EC2 instances. This helps improve the availability and scalability of the application by evenly distributing the workload.
6. Auto Scaling: Auto Scaling is configured to automatically adjust the number of EC2 instances based on the demand. It ensures that the application can handle increased traffic by dynamically scaling the number of instances up or down.

**Facility Pro serverless architecture requirement**

For serverless architecture these are the requirement to host a web application such as WordPress website.

1. Amazon s3 Bucket: S3 bucket is used for hosting static websites.
2. AWS Lambda: lambda is a compute service that provide the developer freedom to run the code without needing to manage the underlying services. It automatically manages the infrastructure that is needed for the application.
3. API Gateway: The API gateway acts as a central gateway for the API, provide feature such deployment and integration with the backend.
4. DynamoDB: DynamoDB provide high performance, scalability for application that require high throughput and low latency.
5. CloudFront: For the content delivery we need CloudFront to deliver content from the nearest availability zone to reduce latency for Facility Pro services.

The main objective of employing serverless architecture is to enhance the following aspects:

1. Reliability: Serverless architectures leverage the distributed nature of AWS services to ensure high availability and fault tolerance. Applications built using serverless components can withstand failures and maintain reliable performance.
2. Security: By utilizing serverless services provided by AWS, security concerns can be mitigated as AWS handles infrastructure security and implements best practices for data protection.
3. Performance: With serverless architecture, applications can benefit from automatic scaling and load balancing capabilities.
4. Cost optimization: Serverless architectures enable efficient resource utilization by dynamically scaling resources based on demand.