



# Implementing Power BI and Migrating to Azure for Optimal Decision-Making

The customer is an Israeli manufacturer of high-quality products for residential, industrial, commercial and office buildings. The products are used in rooms, kitchens, bathrooms, lobbies and more. The company invests heavily in innovation and R&D, manufactures its products in factories in Israel and overseas, and operates offices and sales channels in dozens of countries. The company, which has more than 1,500 employees worldwide, meets strict standards for quality, environmental protection and public health safety. It collaborates with well-known designers around the world.

## The Challenges

The company has various on-premises operational IT systems, supporting their business and operations. These systems process and store large amounts of data for ongoing operations and planning. The company decided to use this quality data and to build a complete data warehouse that would collect data from various local systems and allow their use by the business units for the benefit of BI and advanced analytics. The purpose was to get real-time information, analyze it and make business decisions based on reliable and available information. As a result, the company would undergo digital transformation to streamline processes better and support its business strategy, gaining operational enhancements in various areas.

A data-driven business approach was necessary in all areas connected to management and operations: customers, products, inventory, manufacturing, sales etc. The combination of leveraging existing and future data sources, and the power and scale of the cloud, was key to adopting a comprehensive BI and data management strategy, and digital transformation for meeting current and future needs globally.

## The Solution

The company decided to choose Microsoft Power BI, move the business to the Azure cloud, and work with Ness as the main contractor and integrator.

Ness recommended a solution built on Azure Data Factory (ADF), Microsoft's Extract-Transform-Load (ETL) solution for Azure (Data Factory), and SQL Server – Microsoft's relational database management system. With ADF, Ness could set the schedules for the data transfer and move the data in a web user interface.

The project included setting up the communication infrastructure for Azure, creating the development and production environments, setting up the server infrastructure in Azure, backing up the database and uploading it to the storage account, setting up SQL Server – including the databases on the server, and testing. The process included Refactor SQL on Azure.

Ness decided to use a role-based access control (RBAC) mechanism to prevent overexposure of services and reduce vulnerability from attackers. The RBAC solution gives the company full control in the Azure portal and enables securing the services and assigning users only the actions they need to perform, thus reducing the risks and threats from both inside and outside the organization.

### Ness used built-in roles in the Azure portal, including:

- **Owner** – management of all resources including permissions. The permission is given at the resource group level and to a limited number of users.
- **Reader** – viewing all resources consumed by the resource group without the ability to make changes.
- **Contributor** – managing all resources in Azure except the ability to manage permissions. The permission is given at the resource group level and to a limited number of users.

## Migration to Azure

In the Extraction stage, the relevant data is extracted in an ETL process from the source files at the on-premises operational systems server, installed in the local data center, and copied to an intermediate data warehouse based on Azure Data Lake v2.

In the Transformation stage, the data undergoes changes, such as summarizing data, changing the data structure and making the data ready to load into the target data warehouse environment. For example, data on sales of products, which may be interesting for making business decisions, is summarized according to specific geographical areas.

In the Loading stage, the data that has been transformed is loaded into the central data warehouse in Azure which is based on SQL Database, where the data is processed and displayed using the Power BI platform.

## Implementing BI on Azure

The Power BI-based solution included establishment of services in Azure, including two dedicated resource groups, one for development environment and the other for production environment.

The Network Security Group provides control over network traffic flowing in and out of the services running in Azure. The Azure Virtual Network (VNet) enables the Azure resources to communicate securely with each other, the internet, and on-premises networks.

The Azure Storage Account, which contains the company's Azure Storage data objects, is used to upload files that are extracted from the local on-premises operational systems server to dedicated Azure Data Lake storage, and uploading the files into the systems for analyzing the information. It includes the Azure Data Lake Storage Gen2, which is dedicated to big data analytics, and Azure Blob storage, allowing the storage of large amounts of unstructured data.

A new Windows virtual server was established in Azure, including a managed disk that is used by the development teams to access the SQL Server, including the databases.

There is no downtime of the service in the migration process to Azure.

## Current Status

The infrastructure needed for migrating to the cloud, and the infrastructure in the cloud itself, were fully implemented. In the first phase, a profitability module was implemented, allowing profitability to be measured for products and customers. The module also examines profitability according to various scenarios. The profitability module has already detected various anomalies, allowing the company to make necessary changes.

Two additional modules – sales and logistics – are in advanced development and will be implemented by the end of 2020. Additional modules will be implemented later.