In my current organization, I was involved in a project called the **Borewell Water Consumption Monitoring System**, which aimed to automate and modernize how water usage was tracked across 45 industrial borewells located in different bottling plants. Before this system was implemented, water consumption was recorded manually by field staff. This manual process often resulted in inconsistent data, reporting delays, and a lack of real-time insights, which made it difficult for plant managers to make timely decisions, especially during high-demand periods. Each borewell could pump up to 65 kiloliters of water per hour, so inaccurate tracking had a significant operational and environmental impact.

To solve this problem, we installed **Ultrasonic and Electromagnetic Flow Meters** on each borewell and connected them to **IoT Edge devices**. These edge devices were configured to send telemetry data every minute to **Azure IoT Hub**, which acted as the central data ingestion point. I was responsible for designing and provisioning the entire cloud infrastructure using **Terraform**, so that the deployment of services was automated, repeatable, and consistent across development, test, and production environments. Once data entered the system through the IoT Hub, we used **Azure Stream Analytics** to process and filter the data in real-time. Processed data was then stored in **Azure SQL Database** for structured reporting, and **Azure Data Lake Gen2** for archival and deep analytics.

To visualize the data, I built interactive **Power BI dashboards** that allowed plant managers to monitor daily water usage per borewell, identify abnormal consumption patterns, and compare usage across plants. One of the key features we added was real-time alerting. Using **Azure Functions**, we developed serverless workflows that would automatically trigger notifications if a borewell exceeded the expected usage threshold. These alerts were sent to teams via email and Microsoft Teams using **Power Automate**, enabling quicker responses to potential leaks or over-usage.

Additionally, we integrated **Azure Machine Learning** to forecast future water demand based on historical consumption patterns, plant operations schedule, and environmental data. This predictive insight helped the operations team allocate borewell resources more efficiently and plan preventive maintenance. The infrastructure was designed with strong **network security** in mind: we created separate subnets for IoT, backend processing, and databases within an **Azure Virtual Network (VNet)**, protected by **Network Security Groups (NSGs)** and **Azure Firewall**. Sensitive credentials and connection strings were managed securely using **Azure Key Vault**, and we used **RBAC policies** to ensure only authorized users and services could access critical resources.

From a DevOps perspective, I implemented end-to-end **CI/CD pipelines using Azure DevOps with YAML**, which included infrastructure provisioning via Terraform, deployment of Azure Functions and Stream Analytics jobs, and validation checks. These pipelines helped our development team push changes more frequently and with fewer issues, as everything from infrastructure to code deployment was automated and tested. For monitoring and observability, we used **Azure Monitor**, **Application Insights**, and **Log Analytics** to track telemetry from all services. I set up custom dashboards and alert rules so that any failure in data ingestion, function execution, or resource performance would be detected and flagged immediately.

In parallel with this, we also worked on a **Power and Motor Monitoring System**. We integrated energy meters using the same IoT architecture to collect data on electricity usage. This helped the plant team identify peak usage hours and optimize load distribution, leading to reduced energy costs. I also implemented a **Motor Running Hours Monitoring System** using IoT-based sensors that tracked the operating time of motors. This data was crucial for **predictive maintenance**, allowing us to reduce unnecessary runtime, avoid breakdowns, and save energy.

The overall impact of this project was significant. It replaced a completely manual and error-prone process with a **fully automated, real-time monitoring and alerting system**. It improved resource efficiency, reduced water and energy waste, and gave plant managers deep insights into operations through powerful dashboards and predictive analytics. From a DevOps standpoint, it demonstrated how **cloud automation, infrastructure-as-code, CI/CD, and observability** can come together to support a highly scalable and secure industrial IoT solution.