**IoT Watering System**

## **Overview:**

We would like you to design a system that will be used globally to monitor water pumps for farmers and manage the flowing water at a remote water source.

The idea is that a farmer will have a number of water pumps on their farm that ensures that there is a steady flow of water for irrigation or livestock. As the farms can be quite vast, especially in places like Australia, the system will allow the farmer to monitor that all water pumps are working successfully or provide a notification when they are malfunctioning.

The monitoring device would be a cheap IOT device, that will send information about current status of the pump every 30 minutes. The device is Solar / Battery operated, and it will also be equipped with a 3g modem for transmission of the data, and various other sensors.

It is expected that a farmer will have device between 10 and 1000 devices on their farm, that they can register the device themselves via a website portal. In this portal, they will be able to monitor all of their devices and will also be provided with a dashboard of different metrics, e.g. current water flow. They will be able to control individual devices and set up general rules for the devices, for example, to reduce or increase the flow of water based on the current temperature or humidity. They will also be able to register the telephone and email for SMS alerts.

We expect this service to be very popular around the globe, especially during daylight hours. We expect up to 70,000 users per hour using the service, but that traffic will mainly follow the sun. We would also like to keep running costs to a minimum.

Due to international laws, US data and EU data must be stored in respective continents.

The developers who will be writing the code will rely on your guidance and expertise in the selection of the appropriate languages, please provide such guidelines in the readme, as well as a simple architecture diagram to help them visualize your concept.

If there are some components that cannot be created with the Deployment template, please indicate these clearly, and provide some guidelines on how they should be set up.

The templates should provide recommend sizing for the services, and an estimated running cost. You are free to architect this in any way you wish, using any and all (including beta) services in Azure, but this should be of a production quality level.

----------------------------------------------------------------------------------------------------

# **Key considerations while designing this system:**

* Web Portal to register and un-register the IoT device, visualize metrics in the dashboard, interact with individual devices, set rules on the devices, configure to send SMS alerts.
* Registered device will send data every 30 minutes
* Device – Solar 3g modem
* A farmer can register anywhere between 10-1000 devices
* A dashboard with metrics should be developed with real-time data on the current water follow from the devices
* 70,000 users per hour using the service
* Geo-Availability, Auto scale, System Monitoring
* Keep the cost low

# **Technology considerations:**

Azure IoT Hub:

Ability to connect to billions of IoT devices, bi-directional communication, security, device provisioning service, remote device management.

Stream Analytics:

Develop complex event processing, scalability, pay per job, real-time dashboards, auditing, extend streaming logic, machine learning integration

Power BI:

Turn data into analytics and reports, real-time insights, built-in connectivity with stream analytics, event hubs, machine learning, storage, etc.,

App Services:

Build and host web applications, auto-scaling, high availability, enable automated deployments.

Azure Functions:

Easily build the apps you need using simple, serverless functions that scale to meet demand. Functions supports triggering an event based on an activity in an Azure service.

Azure service SDK’s:

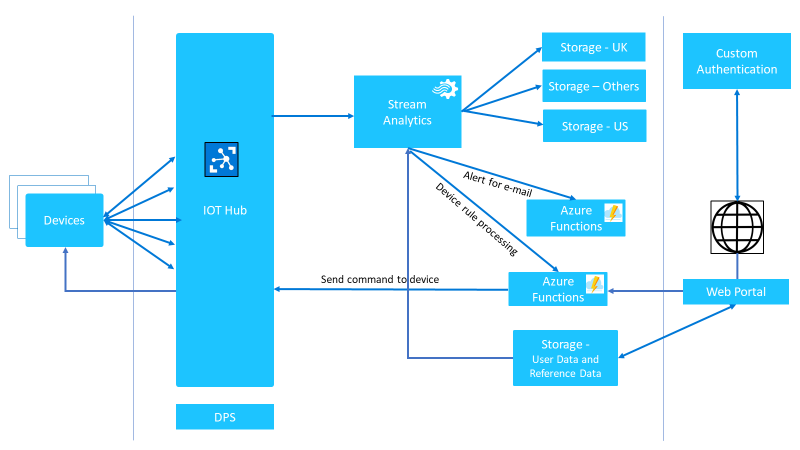
Storage:

Node.js:

Azure Active Directory Services:

Logic App:

**Technical Architecture:**



Capacity Planning:

IoTHub has the capability of handling billions of messages. We are going to use SKU S3 which has the capability to receive from devices up to 6000/sec/unit and the same level for device connections as well. One assumption is that there are 70,000 total users.

There will be 70,000 users per hour using the service. A farmer can register anywhere from 10-1000 devices in the farm.

Mean Proportion is 100, but let’s keep it at 400 devices

(70,000 users \* 400 devices) \* (2 messages / hour) = 56,000,000 messages / hour

56,000,000 \* 24 (1 Day) = 1,344 Million Messages / Day

IoTHub: S3 Plan can handle 300 M / unit / Day

300 M / unit \* 4 = 1,200 Million / Day. So, *S3 with 4 unit can handle the work load.*

Devices to Cloud:

S3 Plan can also capable of handling 6000 messages / sec / unit.

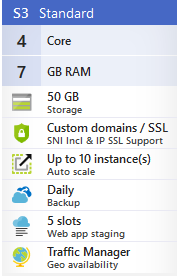
*So*,

6000 \* 60 \* 60 = (21,000,000 / unit) \* 4 = 63,000,000 / hour

According to poison distribution, 27% of the time we may get 2 messages at the same time, 54% of time 1 messages.

**Web Portal:** App Service with S2-Standard plan will suit our needs. It’s elastic and can scale up to 10 instances. We can start with this.

This plan has a Traffic Manager which will take care of redirecting the traffic based on the user’s geological location.



Also, it has 5 slot which we can use for Continuous delivery.

**Storage:** We will be storing US and EU data separately in their respective regions.

It’s a standard plan and local region-specific failover.

For 1344 Million Messages times 2 KB of message size, we need approximately 2300 GB of data storage / month.

**Cost Estimates:** Please find the Excel sheet in Github.