clearscale

aws partner network

Premier Consulting Partner

QMI Manufacturing Leverages the Power of IoT for Its Seismic Sensor Network with Help from ClearScale

QMI

Executive Summary

QMI Manufacturing Inc. is a leading manufacturer of advanced safety systems that detect the presence of gas leaks, water leaks, or seismic vibrations. Their primary focus is on proactive solutions that protect lives and property using technology. QMI Manufacturing makes the world a safer place by protecting people in their homes, hotels, hospitals, schools, and shopping malls, often without people even knowing. They resolve safety and protection issues for elevators, escalators, automatic doors, and rapid transit systems — virtually any place people gather to work or play.

For example, the Vancouver, BC region lies along a major fault line and is overdue for a major seismic event by at least 200 years. Having a fully deployed, interconnected tremor-sensing system in place would potentially allow for early warning by detecting P-Wave tremors and alerting before the arrival of the expected and more serious S-Wave earthquake. ClearScale designed and developed a robust and scalable IoT-centered solution to interconnect all of these devices to realize this goal.

The Challenge

ClearScale, an AWS Premier Consulting Partner, was chosen by QMI Manufacturing as the exclusive partner to assist in the development of an earthquake sensor software product utilizing Internet of Things (IOT) services. ClearScale would also develop a custom Web Application implementing a common GUI for devices, groups, and rules management, as well as deploy the whole system on the AWS platform. Once deployed, a network of these seismic alarms would allow QMI Manufacturing to map and determine where earthquakes occur, even when some of these devices aren't connected to the Internet.

QMI Manufacturing presented this challenge to ClearScale to determine how best to execute this vision. To fully realize this effort, QMI Manufacturing determined that the sensor network needed to operate by deploying and configuring small earthquake sensors throughout the Vancouver area by selling them to factories, schools, office buildings, and residential complexes. These sensors would transmit their data remotely, either individually or networked together, and that data would then be aggregated and analyzed for patterns which could preemptively identify earthquake activity in an area, potentially saving lives through early warning and seismic alarms.

The ClearScale Solution

After thorough discussions and requirements-gathering, ClearScale designed a robust cloud architecture utilizing the AWS IoT infrastructure service, offering resiliency, scalability, and a high value proposition to QMI Manufacturing. Leveraging other AWS services like Lambda and EC2 along with an Angular TypeScript framework and the AWS SDK for JS, ClearScale created a fully functioning IoT data capture mechanism that allows the QMI Manufacturing sensors to transmit the seismic activity information on a regular cadence, where data points like date, time, duration, and acceleration in each axis are captured to be analyzed at a later time. The system further allows for these devices to work together if specified threshholdsx are met, and warn each other of incoming seismic waves, offering proactive warning to neighboring devices.

After analysis, Amazon's IoT Shadow service was identified as the best available technology, and utilized as the common communication bus to allow working with devices with a poor connection to the Internet. To allow for a better user experience, it was determined that a folder tree structure would need to be implemented to allow grouping of devices into folders and subfolders. This provided one of the largest technical challenges for the team. The solution provided by ClearScale included no deep limit on folder trees, and automated statistic calculation for every single folder and subfolder on the fly. This subsystem was implemented on top of Akka framework, the toolkit for building true reactive, highly concurrent, distributed, and resilient message-driven applications. There was no external storage utilized for statistic gathering, only IoT messaging for the messages aggregation into groups and statistic calculations. The system always gathers statistics on every refresh cycle for every single group and subgroup, and losing gathered statistics is no longer a concern — by design, the system will restore all gathered statistics automatically after a failure.

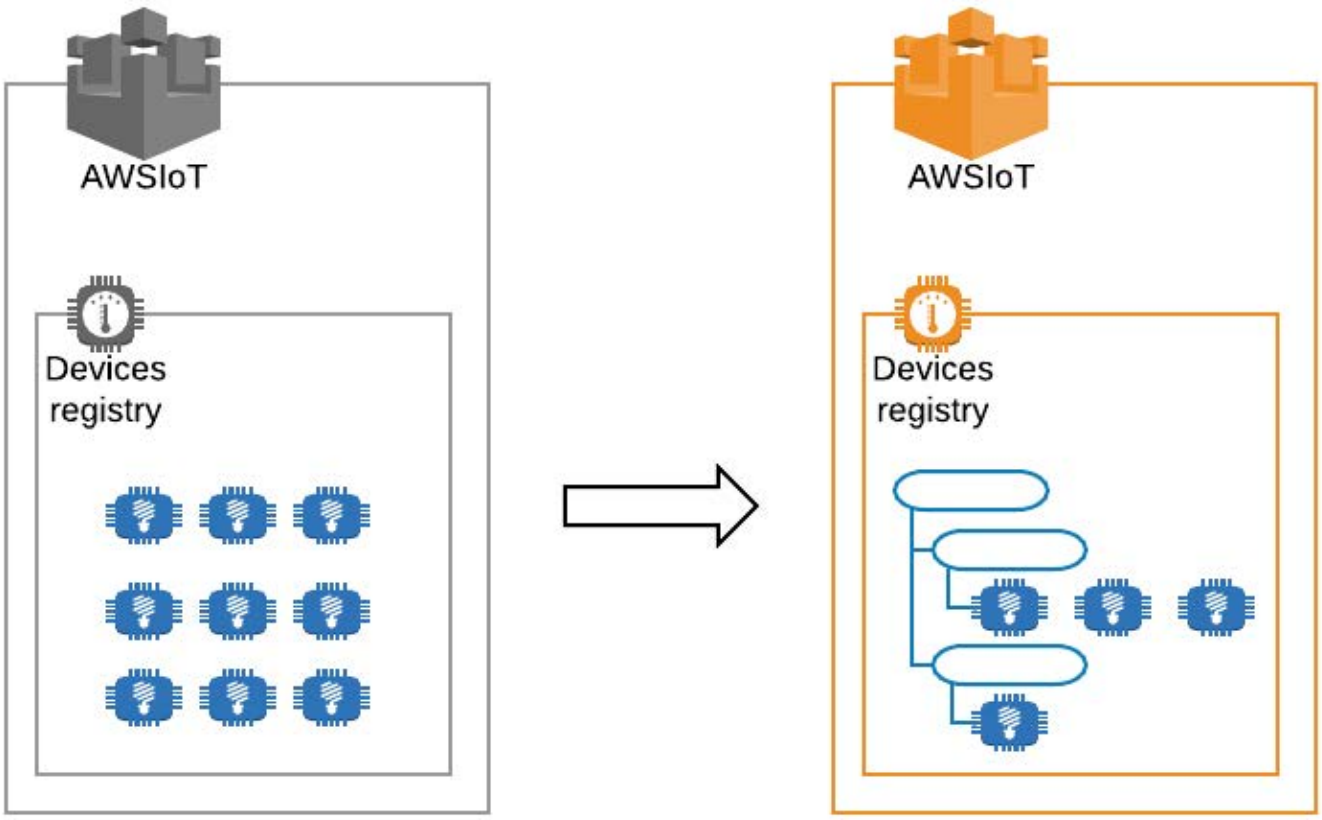
In addition to the backend components, ClearScale provided a robust user interface for the application that provides users with the ability to manage devices and groups of devices. This application was built with modern technologies like Angular, RxJs, HTML5, WebSockets, and more. Utilization of framework libraries expedited development while producing incredible speed and functionality.

AWS IoT Implementation Details

Device Grouping

One of the project requirements was device grouping into folders (2+ levels deep). AWS IoT currently doesn't support device grouping and mainly provides a high performance communication bus for the devices.

ClearScale has implemented this requirement as an external module for AWS IoT, which allows device grouping into a tree structure without restriction on subgroup level depth. No additional databases were used for storing the group information — all information was stored inside AWS IoT device registry (at no additional cost).

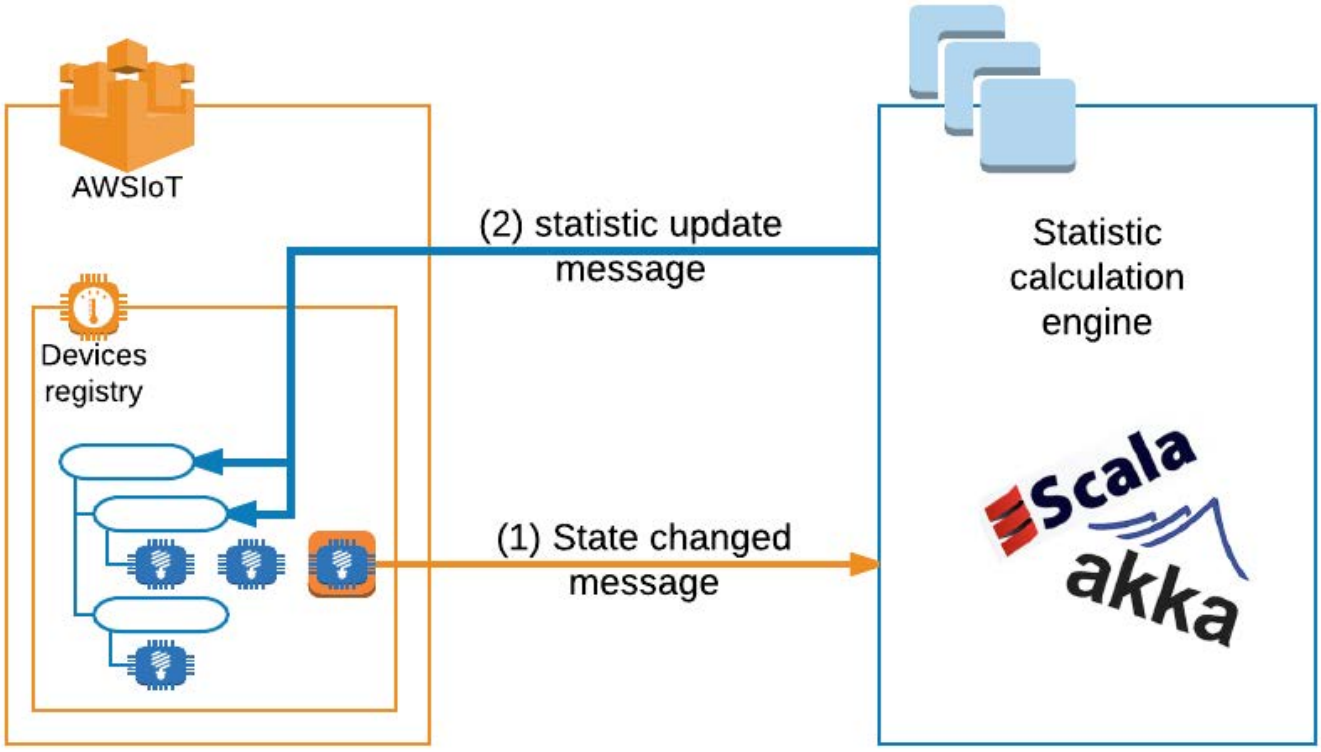


Real-Time Statistics

- Functionality for statistics calculation was the next important piece that was created on top of the device grouping. ClearScale developed this based on the client's requirement that the statistics engine provide summarized group statistics covering all devices.
- The statistics should be available in real-time (updated every 5 seconds on devices).

In full production, where millions of devices are running simultaneously, there is no easy way to gather a statistics summary from all devices on the fly, because it takes a very long time. For that reason, ClearScale has chosen to implement an incremental statistics update per every state change, to process this requirement in pieces.

ClearScale has developed a custom application with Akka which connects to every device and reacts to every state by sending messages to AWS IoT.



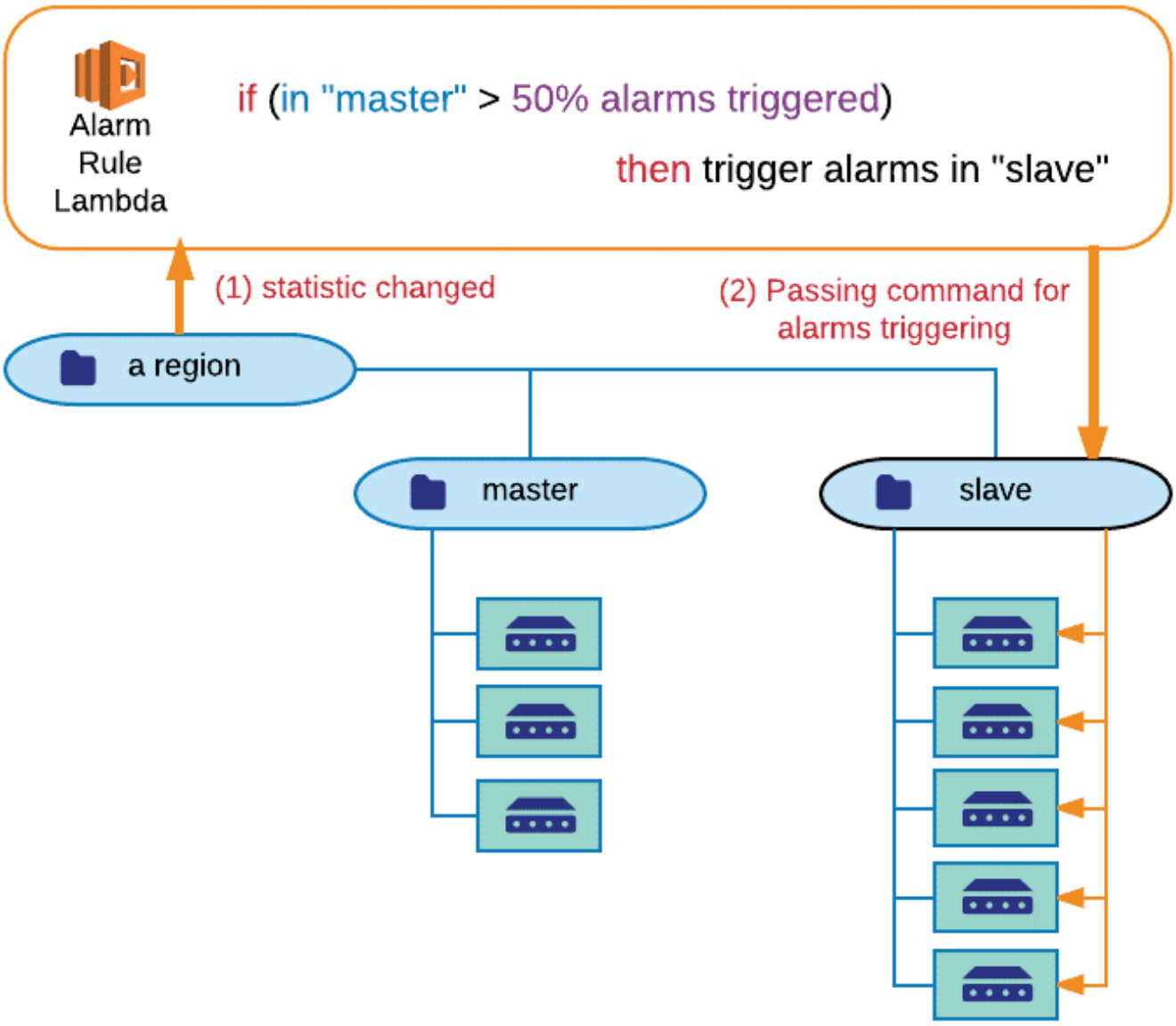
Another AWS component ClearScale utilized was AWS IoT Shadow Service. This service makes it possible to work with devices which are in offline mode without implementing a special backend layer.

In our case, no DB was used for statistics storage — rather all device statistics are stored inside the IoT Shadow Service. This has reduced the cost of the system and made it possible to integrate other applications or AWS services with this statistic engine by using any standard AWS integration with IoT (standard IoT Rules/AWS Lambda functions/AWS SDK).

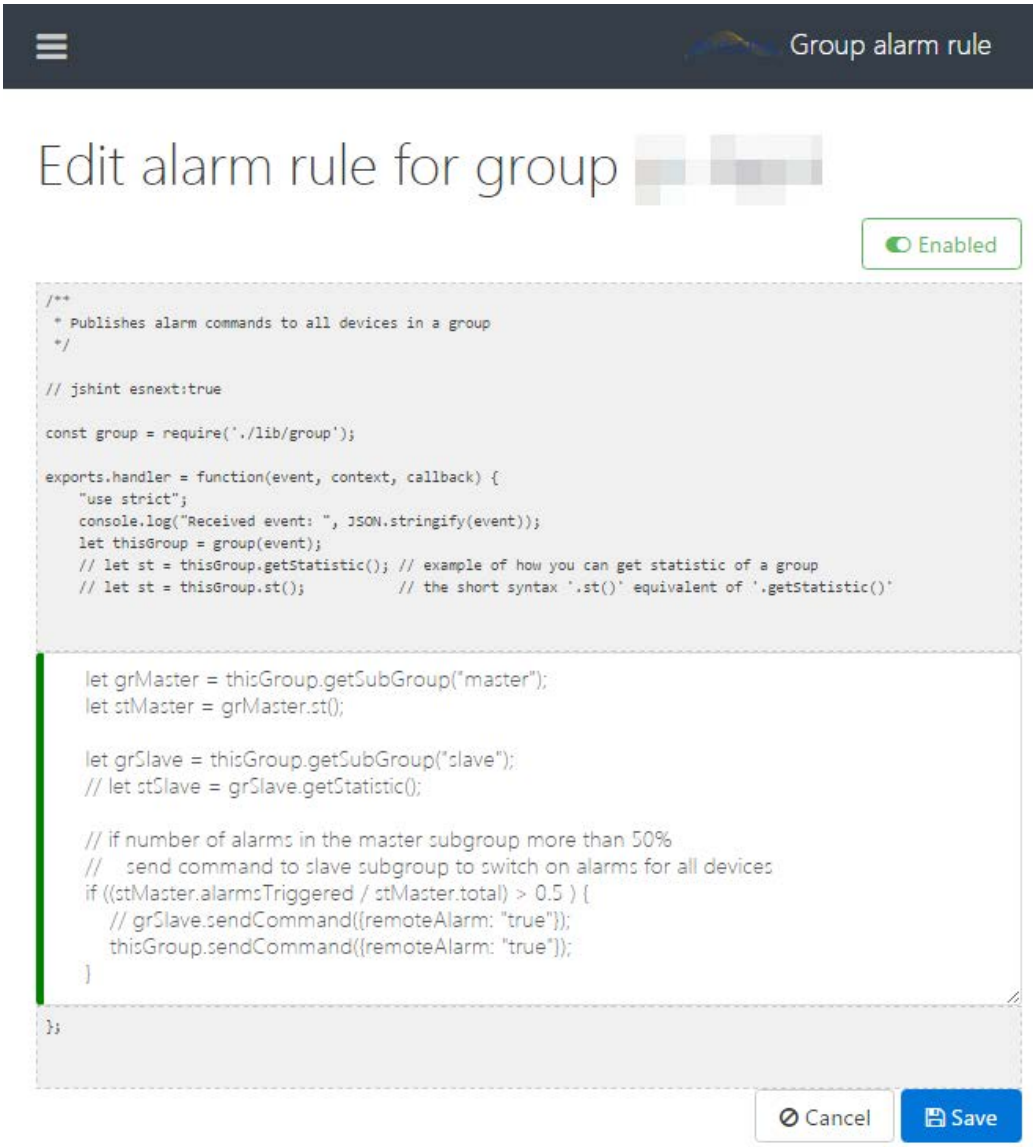
Alarm Rules Engine

Built on top of the other previously mentioned features (device grouping and statistics calculation) is the Alarm Rules Engine (ARE). The ARE system provides the ability to set up rules based on gathered statistics like forcing devices in a group into an alarm mode when that group has triggered a specific number of alarms (for example > 50%).

Under the hood, ClearScale has utilized AWS Lambda and AWS IoT messaging rules engine to make this work with virtually no additional resource usage.



Alarm rules compiled with Node.js Lambda functions were deployed to trigger alarms based on the statistics changes of a group. Setting up and deploying alarm rules requires technical knowledge, so ClearScale created the Alarm Rules Editor as a part of the application to supplement the system. All technical parts of the rules management (compilation, deployment, AWS access control system setup, IoT rules setup, etc.) were automated to minimize system maintenance.



Notifications Engine

The notifications engine was implemented as a part of the solution. Based on AWS SNS and built on top of devices grouping, the notification engine reacts to specific events and notifies authorized users. Notifications delivery is routed via two separate channels: SMS messages and Emails.

To prevent a huge amount of notifications (the system can work with thousands of device groups and millions of devices), ClearScale has implemented a filtering system based on specific statistics monitoring, grouping messages on device-groups, and smart time-frame algorithms.

Automatic Device Registration

Since QMI Manufacturing is going to use millions of devices, it's extremely important to make the device registration process secure and user friendly. To accomplish this, ClearScale utilized AWS IoT approach called "Registration on demand" based on X.509 certificates, Lambda functions, and more:

- Protocol agreement
- X.509 certificate fields agreement
- Certificates generator
- Root certificate
- Documentation

With this feature implemented, devices with properly generated and signed certificates can connect to the system without any special registration process. During the first connection new devices are automatically placed into a special device-group, and can be moved to a specific region group by the admin user (using Management Page on the Web Portal).

Web Management Portal Implementation Details

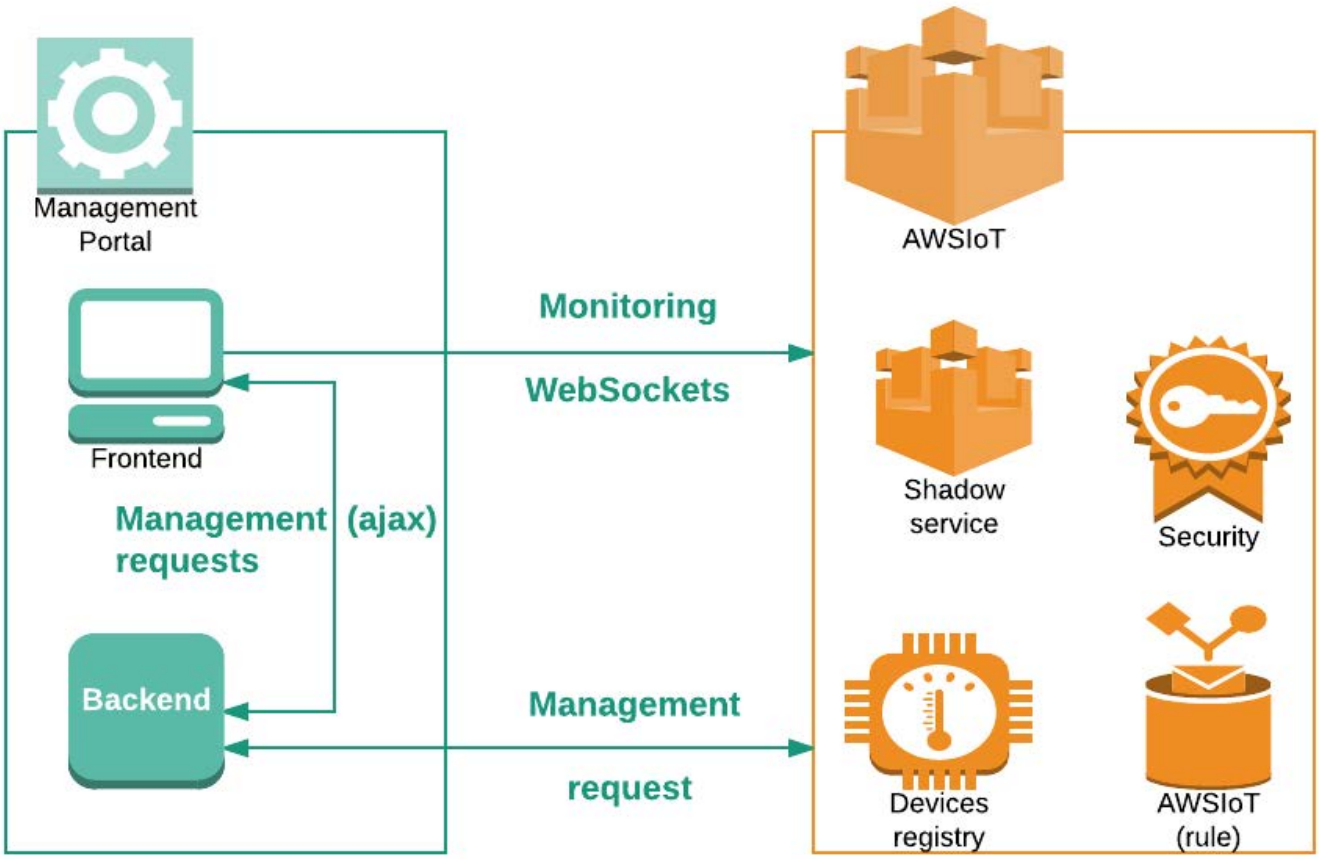
Once the initial design, implementation, and configuration of the IoT solution within AWS had been completed, ClearScale then undertook a development effort to build out a web portal and graphic user interface for system administrators to setup, manage, and monitor devices, manage users and roles, and set up alarm criteria, as well as manage the grouping of sensor devices geographically. This approach provided an easy-to-use solution to the management of a complex and ever-changing sensor landscape as it grows organically over time.

Technology Used

- Web application was built using the following framework components:
  - Java 8
  - Stateless (JWT)
  - Spring family (DI, MVC, Security, Tracking, and so on)
  - Guava
  - EhCache
  - AWS Java SDK
- Angular-based fronted applications:
  - Node.js (for development only)
  - TypeScript
  - Bootstrap
  - RxJs
  - WebPack
  - AWS JS SDK

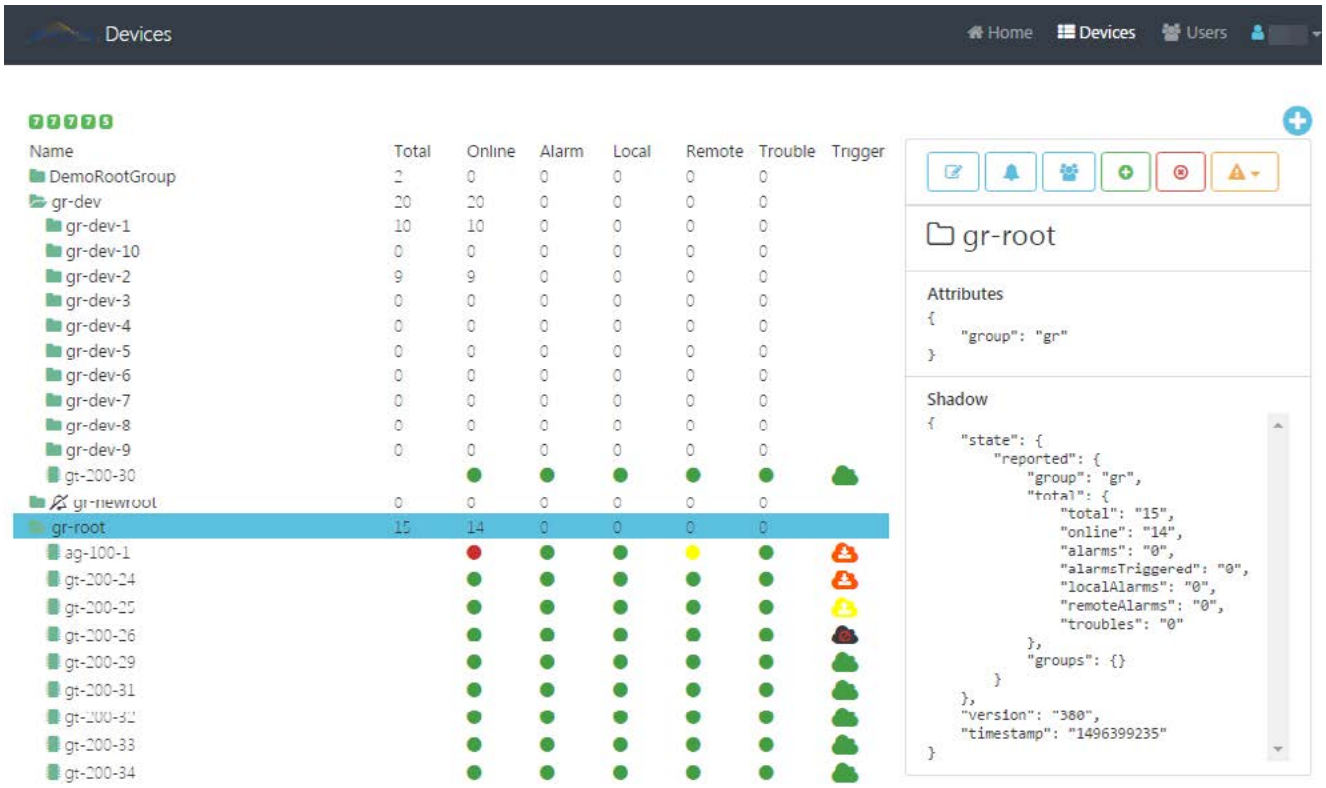
Direct IoT Connection

Since AWS IoT supports Web socket connection, the web portal doesn't need to use its own backend as proxy server for connection to IoT. The connection is obtained via client web browser to IoT directly. This approach ultimately reduces requirements for backend servers, because almost all traffic is generated between AWS IoT and the web browser (instead of a standard connection to its own backend). The website backend has its own API, but only for administrative functions, like user management.



Devices Monitoring and Management

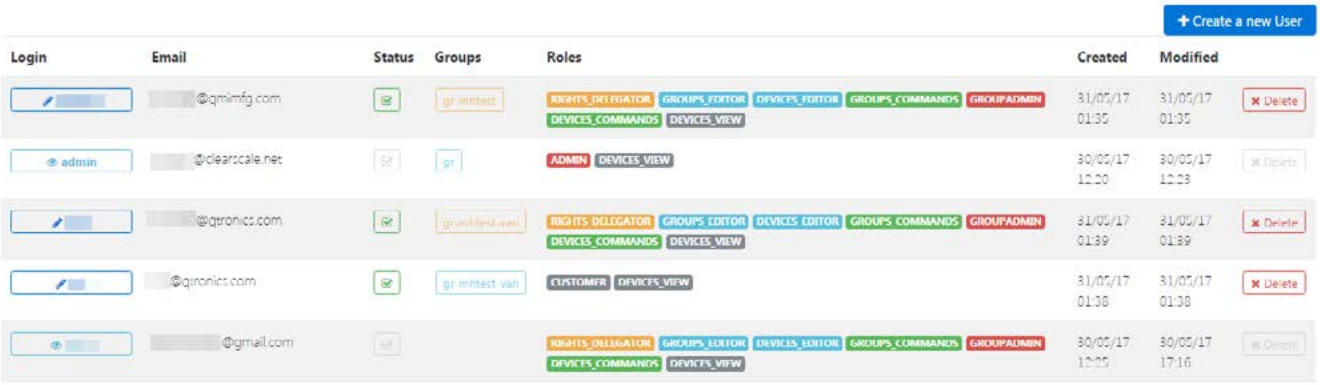
Using a direct connection to AWS IoT (through AWS JS SDK and WebSockets), a device monitoring dashboard was created in the Web Portal. The dashboard displays statistics for every single loaded group and device in real-time, and allows management of the entire device-groups tree.



AWS Cognito Service as a Single User Database

All Web Portal users are managed via AWS Cognito bundles, the service created especially for user management. The main reason to use Cognito was the fact that AWS IoT supports connections via WebSocket with authentication through Cognito service.

Under the hood of the database, ClearScale has implemented a full integration of AWS Cognito with Web Portal's user admin interface, including AWS AIM Roles/policy management, linking with AWS Identity, linking with IoT access rules, etc.



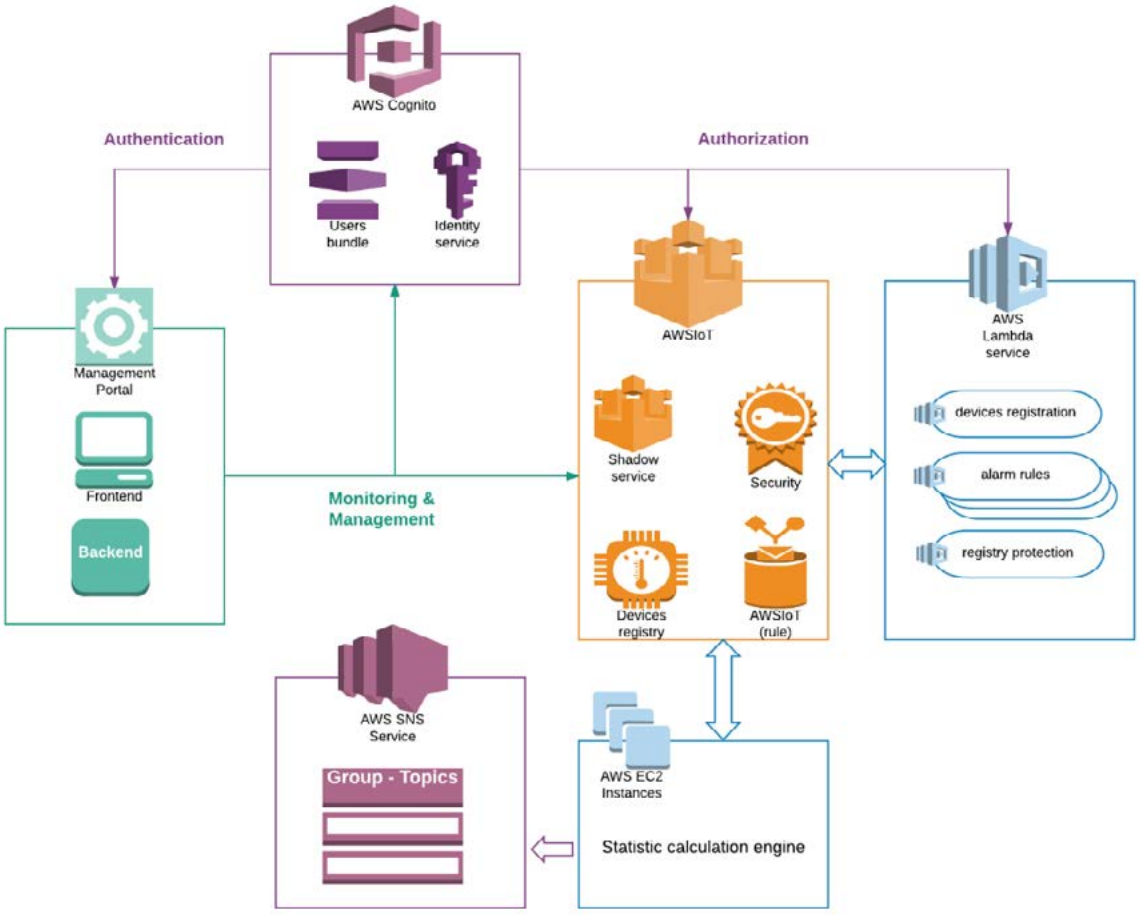
Multi Customer Access Model

The web portal was built utilizing modern technologies like Angular, RxJs, HTML5, WebSockets, and so on. As far as monitoring and management of the system working directly with IoT from a web browser, the security engine was implemented on top of AWS Cloud infrastructure using AWS IoT security rules, AWS Cognito Users, and AIM roles to protect connections only for authorized users with authentications control, which are based on user roles and granted rights to devices groups.

QMI Manufacturing needed the ability to use the system with multiple independent customers, so the access control system takes care of customer isolation. Based on the same tree structure of device groups, every customer has their own "root devices group" ability to create their own devices' hierarchy and manage the users to work with and monitor the devices. This hybrid multi-tenant solution allows for alarms and triggers to be regional, while user access to devices remain customer-centered.

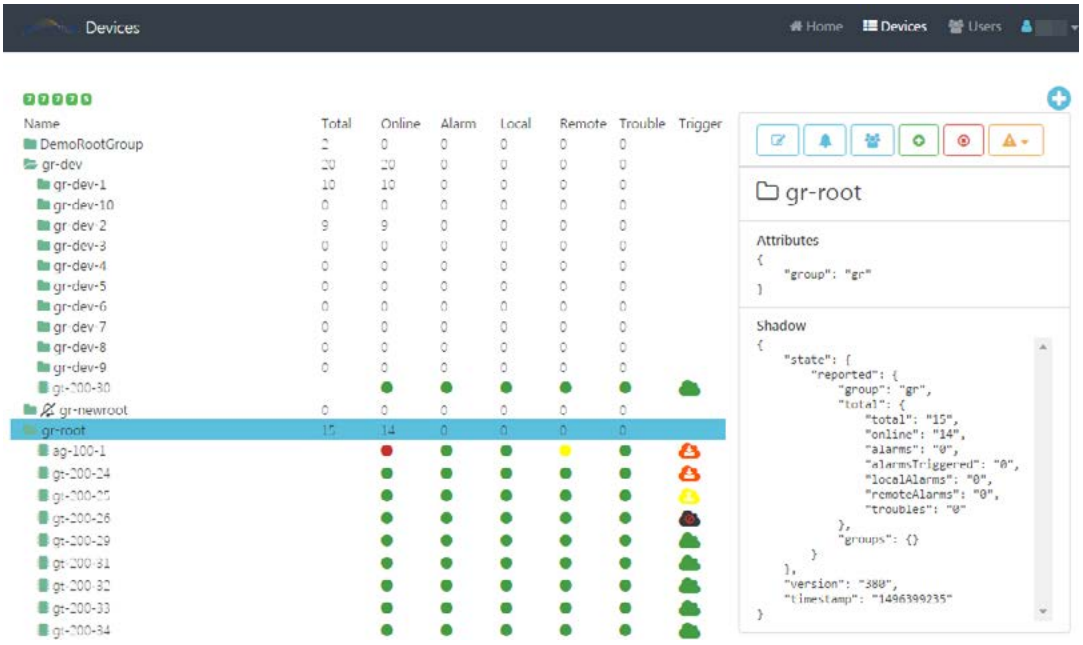
Also, self-management protection (as part of best practice of any user access management system implementation) was implemented, so an administrator can't manage himself or another administrator of the same access level. The root administrator can be managed/created using AWS Web Console (Cognito) only.

The Common System Architecture Diagram

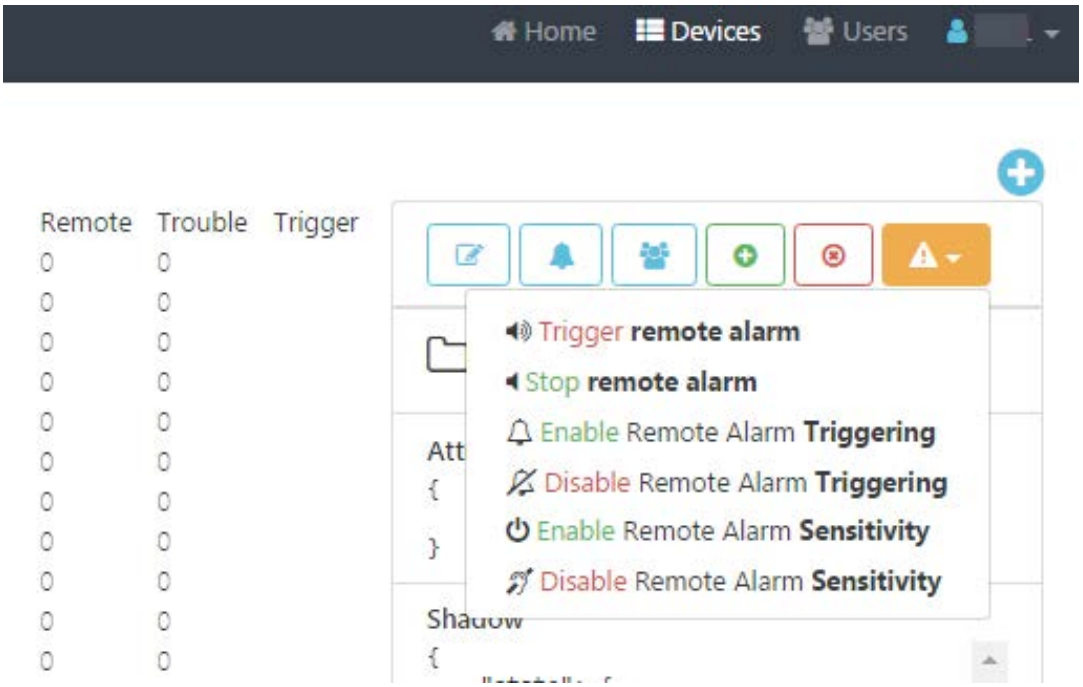


Web Portal Screenshots

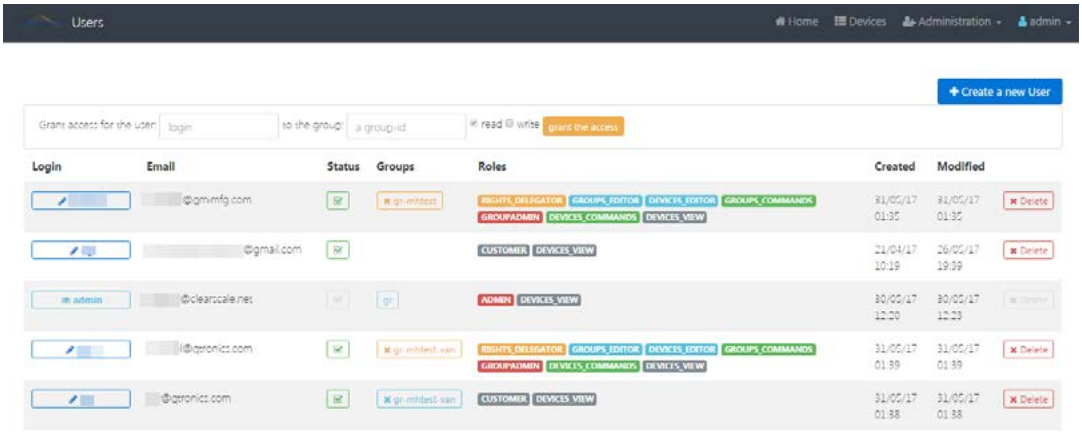
Devices page



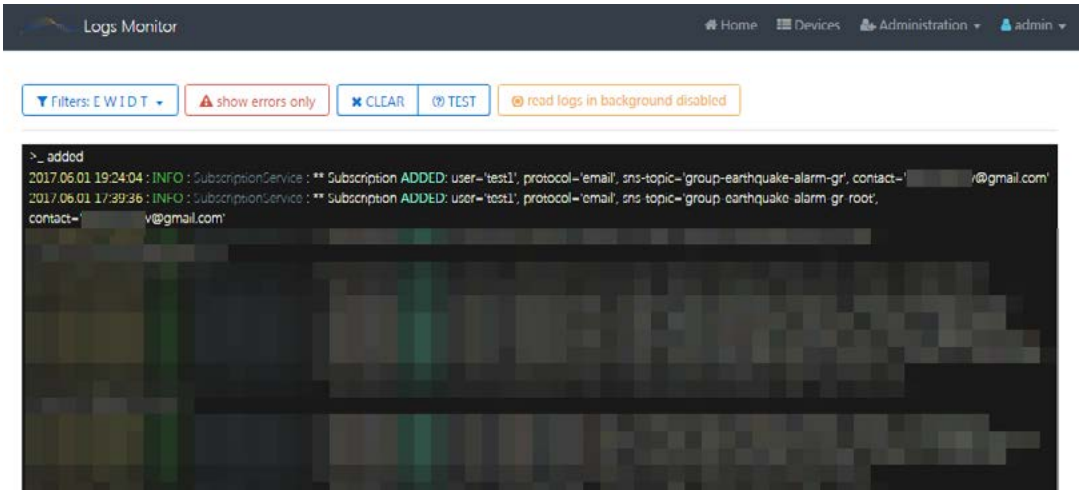
Commands (Device Management) Menu



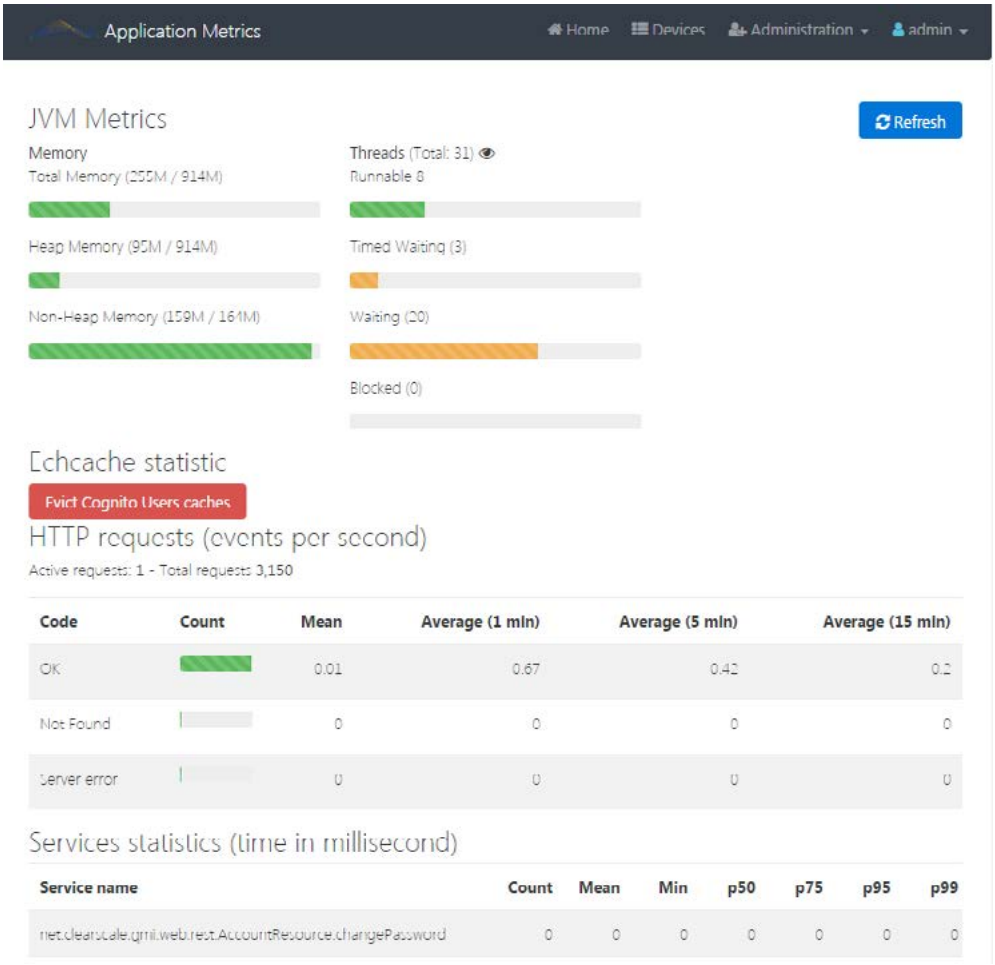
User Management Menu



Administration Area: Online Logging Monitor with Filtering



Administration Area: Backend Metrics Monitoring Page



Value Created

On their own, the QMI Manufacturing earthquake sensors are simply data-gathering devices; they are not fully capable of understanding the significance of how the data from each sensor fits into a larger picture of earthquake activity in the region. By leveraging AWS IoT services and deploying the solution on the AWS platform, ClearScale was able to provide a backbone to the entire sensor network and allow the data to be aggregated and analyzed efficiently in a centralized manner.

In addition, the Web application with user interface that was delivered to QMI Manufacturing gave the company the ability to streamline its management of the devices that would have been impractical to execute had the individual sensors not been aggregating the data in a single IoT location. This also provided a market advantage to QMI Manufacturing by giving them the opportunity to showcase how a widely distributed sensor network can provide detailed temblor data which can then be analyzed and used for early-warning scenarios and, in turn, provide a competitive market advantage and greater market saturation.