Microsoft Cloud Workshop

Serverless architecture

Whiteboard design session trainer guide

January 2018

Information in this document, including URL and other Internet Web site references, is subject to change without notice. Unless otherwise noted, the example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted herein are fictitious, and no association with any real company, organization, product, domain name, e-mail address, logo, person, place or event is intended or should be inferred. Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

The names of manufacturers, products, or URLs are provided for informational purposes only, and Microsoft makes no representations and warranties, either expressed, implied, or statutory, regarding these manufacturers or the use of the products with any Microsoft technologies. The inclusion of a manufacturer or product does not imply endorsement of Microsoft of the manufacturer or product. Links may be provided to third-party sites. Such sites are not under the control of Microsoft and Microsoft is not responsible for the contents of any linked site or any link contained in a linked site, or any changes or updates to such sites. Microsoft is not responsible for webcasting or any other form of transmission received from any linked site. Microsoft is providing these links to you only as a convenience, and the inclusion of any link does not imply endorsement of Microsoft of the site or the products contained therein.

Contents

Serverless architecture whiteboard design session student guide

Abstract and learning objectives

Setup and configure a serverless architecture within Azure using a combination of Azure Functions, Logic Apps, Event Grid, Cosmos DB, and Azure Storage. The focus is on removing server management from the equation, breaking down the solution into smaller components that are individually scalable, and allowing the customer to only pay for what they use.

Learning Objectives:

- Use a series of Azure Functions that independently scale and break down business logic into discrete components.
- Use computer vision algorithms within an Azure Function to accurately detect license plates in car images at scale.
- Provision and use Cosmos DB as a highly available NoSQL data store for processed data.
- Create a Logic App that contains a workflow to export processed license plates and conditionally send alerts based on successful or unsuccessful operation.
- Use App Insights to monitor the serverless topology, observing how well the solution scales when under load.
- Implement a Continuous Deployment DevOps process to automatically publish changes to Function Apps.

Step 1: Review the customer case study

Outcome

Analyze your customer's needs.

Facilitator/subject matter expert (SME) presentation of customer case study

Timeframe: 15 minutes

Directions: With all participants in the session, the facilitator/SME presents an overview of the customer case study along with technical tips.

1. Meet your table participants and trainer.
2. Read all of the directions for Steps 1–3 in the Student Guide.
3. As a table team, review the following customer case study.

Customer situation

Litware, Inc. was founded in 2011 in Houston, Texas, and provides custom software development solutions for a number of clients. In addition to custom software development, they also have developed a financial billing and payment suite of software aimed at several vertical markets, from e-commerce to medical, financial services. They have recently added toll road booth management, as a new market opportunity opened to handle vehicle tracking and toll billing near their home office. Since this new business venture was a minor addition to their impressive portfolio of billing services, they have not dedicated significant resources to the vehicle processing portion of their custom-built TollBooth software suite. The most feature-rich component of this software suite is their existing payment management system that has been expanded to send bills to drivers after passing through any number of the managed toll booths. Included in the bill are a date/time stamp, toll booth location, and a photo of the vehicle as it passed through the booth.

Because so few resources were applied to the TollBooth software, which was meant to handle just a handful of local toll booths, Litware has been using a manual process to identify license plates and send that data to their billing software. As a car passes through a toll booth, a medium resolution image is taken of the car to identify its license plate numbers/characters which will ultimately be used to look up and bill the customer. Currently, they periodically package and send those images to a 3rd party vendor, who manually identifies the license plate numbers and sends the list back to Litware when they are done. At this point, Litware collects batches of 1,000 transactions, saves the information to a CSV file hosted by an FTP server, where their downstream accounting system extracts the license plate information and bills the customer.

Litware has recently been awarded a large, yet unexpected contract to manage toll booths across most of the state, resulting in a 2500% increase in coverage, and is in talks with Oklahoma and New Mexico to provide toll booth services in those states as well. Despite the obvious benefits of such rapid growth, the company is concerned that they will be unable to meet the demand that comes with it. They are confident that their billing software can handle the load, as it has been the primary focus of development from the start, and has expanded into other markets, proving its ability to handle large-scale transactions and data processing. However, Litware is concerned about how rapidly they can automate the license plate processing portion of their TollBooth infrastructure, while ensuring that the automated solution can scale to meet demand, particularly during unexpected spikes in traffic.

“What we need is a lightweight, yet powerful method that quickly pulls in vehicle photos as they are uploaded, and intelligently detect the license plate numbers, all while efficiently handling spikes in traffic,” says Abby Burris, CIO, Litware, Inc. “Most importantly, we do not want to manage long-lived application instances, we want to minimize our cost during slow traffic periods, and we need something our developers can quickly integrate into our existing infrastructure without a lot of training. Our primary goal is to rapidly replace this manual processing pipeline while continuing to devote our development resources to our core billing platform services.”

Abby went on to say that she has been following the relatively new serverless computing movement, and believes that the benefits serverless brings a good match for what they are hoping to achieve in this project. The fewer infrastructure responsibilities for the already maxed out IT team, the better. However, she is not sure whether it is possible to locally develop the serverless components and automate the deployment process using CI/CD DevOps practices like they can with their more traditional web applications.

Since Litware does not have any machine learning experts or data scientists on staff, they would like to know their options for using a ready-made machine learning service that can perform the license plate recognition task on the photos. They would prefer to go this route, rather than to train their staff to properly create and train advanced machine learning models, then having to incur the cost of hosting their own machine learning service for conducting this one task.

Litware wants to store captured vehicle photos in cloud storage for retrieval via custom web and mobile applications. These photos will need to be accessible by the downstream billing service for inclusion on the customer bills. Also, any photos containing license plates that could not be automatically detected will need to be marked as such and accessed later on for manual validation. Similarly, as photos are successfully processed for license plate detection, the plate information needs to be saved to a database, along with the capture date/time and tollbooth Id. Litware has a customer service department who can monitor the queue of photos marked for manual validation, and enter the license plates into a web-based form so they can be exported along with the automatically processed license plate data.

The process to export license plate data also needs to be automated. Litware would like an automated workflow that runs on a regular interval to extract new license plate data since the last export and saves it in a CSV file that gets ingested by the billing software. They already have the CSV ingest process automated, so no changes are required beyond saving the file. Their FTP server would need to be modified to point to the cloud storage container instead of its local file system, which is a simple process that is out of scope for the automation task. The export interval should be set to one hour, but be flexible to increasing or decreasing the interval as needed. This interval is based on the automated file ingest process used by the billing system.

Customer service has requested that an alert email should be sent to a specific monitoring address if at any point the automated export does not complete due to no data. Given the export interval and the average number of vehicles that pass through the toll booths during any given hour, having no data to export would be the exception, not the rule. The alert would give them the peace of mind that they could go through internal support channels to investigate the license processing pipeline to address any issues promptly, without being inundated by too many unnecessary alert notifications. They are using Office 365 for their email services.

In addition to the email alert notifications, Litware would like to have a centralized monitoring dashboard they can use to watch the automated process in real time and drill down into historical telemetry later on if needed. This dashboard will help them keep an eye on the various Azure components, watching for any bottlenecks or weak points in their overall solution. The monitoring dashboard should also allow them to add custom alert notifications that get sent to IT staff if anything goes wrong.

“Our directors want to see where we can take the notion of a serverless architecture, and see if there truly are long-term performance and cost benefits,” says Burris. “With the unexpected windfall of the toll booths contract, they want to make sure we have a tested strategy we can fall back on in the future when our IT and development teams are called upon once again to achieve the impossible.”

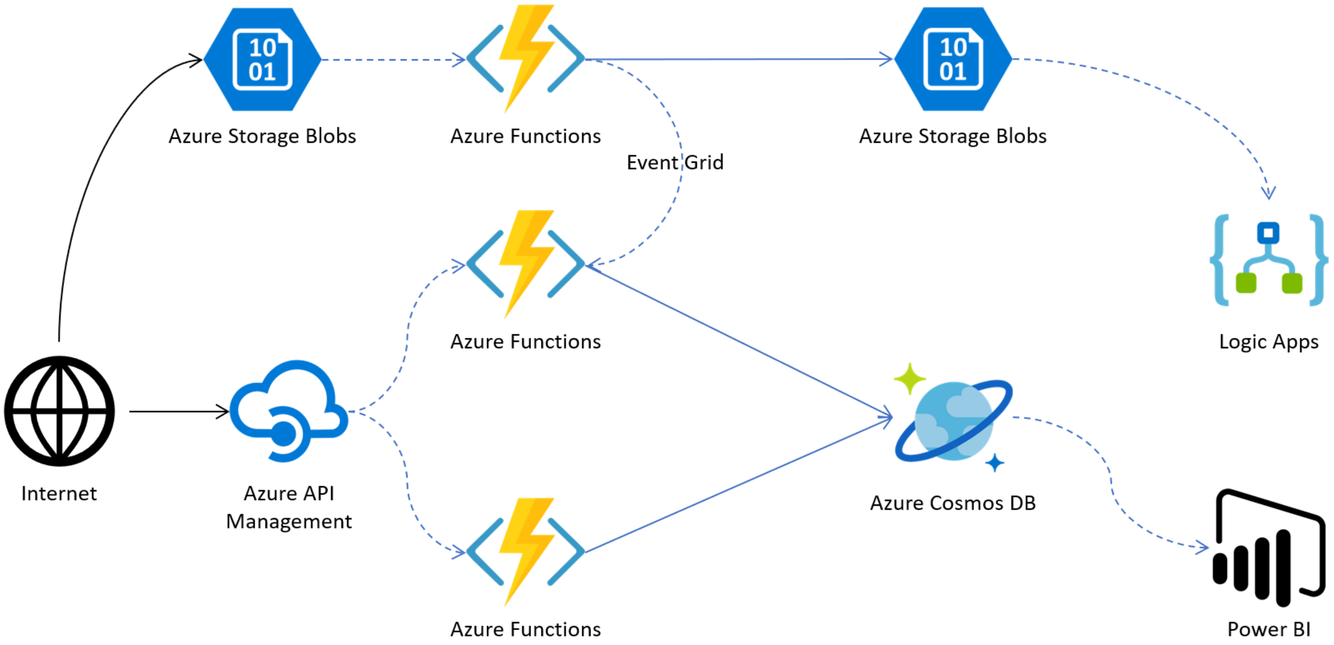
Customer needs

1. Replace manual process with a reliable, automated solution using serverless components.
2. Take advantage of a machine learning service that would allow them to accurately detect license plate numbers without needing artificial intelligence expertise.
3. Mechanism for manually entering license plate images that could not be processed.
4. Have a solution that can scale to any number of cars that pass through all toll booths, handling unforeseen traffic conditions that cause unexpected spikes in processed images.
5. Establish an automated workflow that periodically exports processed license plate data on a regular interval, and sends an alert email when no items are exported.
6. Would like to locally develop the serverless components and establish an automated deployment pipeline from source control.
7. Use a monitoring dashboard that can provide a real-time view of serverless components, historical telemetry data for deeper analysis, and supports custom alerts.

Customer objections

1. We are concerned about how individual serverless components will be able to “talk” to each other and reliably pass messages through the pipeline.
2. Will a serverless architecture that has the capacity to infinitely scale put us at risk for huge monthly bills?
3. How do we make sure that erroneous image processing does not make certain toll bills fall through the cracks or, even worse, send a bill to the wrong person?

Infographic for common scenarios



Step 2: Design a proof of concept solution

Outcome

Prepare to present a solution to the target customer audience in a 15-minute chalk-talk format.

Timeframe: 60 minutes

Business needs

With all participants at your table, answer the following questions and list the answers on a flip chart.

1. To whom should you present this solution? Who is your target customer audience? Who are the decision makers?
2. What customer, business needs should your solution address?

Design

With all participants at your table, respond to the following questions on a flip chart.

High-level architecture

1. Without getting into the details (the following sections will address the particular details), diagram your initial vision for handling the top-level requirements for the license plate processing serverless components, OCR capabilities, data export workflow, and monitoring plus DevOps.

License plate processing serverless components

1. Which Azure messaging service would you recommend using to orchestrate event-driven activities between the serverless components?
2. What Azure service would you suggest Litware use to execute custom business logic code when an event is triggered?
3. Which pricing tier for the service would you recommend that would automatically scale to handle demand while charging only for work that was performed?
4. How do you ensure that downstream components, such as machine learning APIs, databases, and file stores, are not overloaded by the potential high load created when your serverless components dynamically scale?
5. What Azure service would you recommend for storing the license plate data? Consider options that automatically scale to meet demand, and offer bindings to other serverless components that simplify connecting to and storing data within the data store.

License plate OCR

1. What service would you recommend Litware use to conduct license plate object character recognition (OCR) processing to extract the license plate number from each photo as it enters the system?
2. How would you integrate the OCR service to your license plate processing flow?

Data export workflow

1. What Azure service would you recommend to create an automated workflow that runs on a regular interval to export processed license plate data and send alerts as needed?
2. Which other services would you integrate into your workflow?

Monitoring and DevOps

1. What tools and services would you recommend Litware use to develop the serverless components locally, synchronize with a source code repository, and implement continuous deployment?
2. How would you monitor all the executing serverless components in real time from a single dashboard?
3. Does your monitoring solution support exploring historical telemetry and configuring alerts?

Prepare

With all participants at your table:

1. Identify any customer needs that are not addressed with the proposed solution.
2. Identify the benefits of your solution.
3. Determine how you will respond to the customer’s objections.

Prepare a 15-minute chalk-talk style presentation to the customer.

Step 3: Present the solution

Outcome

Present a solution to the target customer audience in a 15-minute chalk-talk format.

Presentation

Timeframe: 30 minutes

Directions

1. Pair with another table.
2. One table is the Microsoft team, and the other table is the customer.
3. The Microsoft team presents their proposed solution to the customer.
4. The customer makes one of the objections from the list of objections.
5. The Microsoft team responds to the objection.
6. The customer team gives feedback to the Microsoft team.
7. Tables switch roles and repeat Steps 2–6.

Wrap-up

Timeframe: 15 minutes

- Tables reconvene with the larger group to hear an SME share the preferred solution for the case study.

Additional references

Item	Description	Links
Azure Functions	Introduction to Azure Functions	https://docs.microsoft.com/en-us/azure/azure-functions/functions-overview
Logic Apps	What are Logic Apps?	https://docs.microsoft.com/en-us/azure/logic-apps/logic-apps-what-are-logic-apps
Azure Cosmos DB	About Azure Cosmos DB	https://docs.microsoft.com/en-us/azure/cosmos-db/introduction
Messaging service	Choose between Azure services that deliver messages	https://docs.microsoft.com/en-us/azure/event-grid/compare-messaging-services
Event Grid	Introduction to Azure Event Grid	https://docs.microsoft.com/en-us/azure/event-grid/overview
Application Insights	Monitor Azure Functions using Application Insights	https://docs.microsoft.com/en-us/azure/azure-functions/functions-monitoring
Logic Apps	Call Azure Functions from logic apps	https://docs.microsoft.com/en-us/azure/logic-apps/logic-apps-azure-functions#call-azure-functions-from-logic-apps
Azure Cosmos DB + Azure Functions	Azure Cosmos DB: Serverless database computing using Azure Functions	https://docs.microsoft.com/en-us/azure/cosmos-db/serverless-computing-database
Azure Functions	Continuous deployment of Azure Functions	https://docs.microsoft.com/en-us/azure/azure-functions/functions-continuous-deployment
Azure Functions	Code and test Azure Functions locally	https://docs.microsoft.com/en-us/azure/azure-functions/functions-run-local