Serverless architecture - Whiteboard design session student guide

November 2023

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.

Nov-2023 Updates Christian Nagel

Contents

Serverless architecture whiteboard design session student guide

Abstract and learning objectives

In this whiteboard design session, you work with a group to design a solution for processing vehicle photos in near real-time, as they are uploaded to a data lake, using serverless technologies on Azure. The license plate data must be extracted and stored in a highly available NoSQL data store for exporting. The data export process is orchestrated by a serverless Azure component that coordinates exporting new license plate data to file storage and sending notifications as needed. You will also configure a Continuous Deployment process to publish new changes to Function Apps automatically. Finally, the entire processing pipeline will need to be monitored, with particular attention paid to components scaling to meet processing demand.

At the end of this whiteboard design session, you will have greater insight into how best to take advantage of serverless architectures. You will understand better how to design highly scalable and cost-effective solutions that require very little code and virtually no infrastructure compared to traditional hosted web applications and services.

Step 1: Review the customer case study

Outcome

Analyze your customer's needs.

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 team members and trainer.
2. Read all directions for steps 1-3 in the student guide.
3. As a team, review the following customer case study.

Customer situation

Contoso Ltd. was founded in 2011 in Houston, Texas, and provides custom software development solutions for many clients. In addition to custom software development, they have also developed a financial billing and payment suite of software aimed at several vertical markets, from e-commerce to medical and financial services. They have recently added toll road booth management as a new market opportunity to handle vehicle tracking and toll billing near their home office. Since this new business venture was a minor addition to their impressive billing services portfolio, 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. The bill includes a date/time stamp, toll booth location, and a photo of the vehicle as it passed through the booth.

Because Contoso applied so few resources to the TollBooth software, which handles just a handful of local toll booths, they have been using a manual process to identify license plates and send that data to their billing software. As a vehicle passes through a toll booth, a medium resolution image is taken to determine its license plate numbers/characters, which Contoso ultimately uses to look up and bill the customer. They periodically package and send those images to a third-party vendor, who manually identifies the license plate numbers and sends the list back to Contoso when they are done. At this point, Contoso 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.

Contoso has recently been awarded a large but unexpected contract to manage toll booths across most of the state of Texas, resulting in a 2500% increase in coverage. Additionally, it is in talks with Oklahoma and New Mexico to provide toll booth services in those states. Despite the obvious benefits of such rapid growth, the company is concerned that it 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, Contoso 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 traffic spikes.

"What we need is a lightweight, 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, Contoso Ltd. "Most importantly, we do not want to manage long-lived application instances. We want to minimize our cost during slow traffic periods and 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 says that she has been following the relatively new serverless computing movement and believes that serverless architecture's benefits bring a good match for what they hope to achieve in this project. The fewer infrastructure responsibilities for the already maxed out IT team, the better. However, she is unsure 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.

Contoso does not have any machine learning experts or data scientists on staff, so they want to understand better their options for using a ready-made machine learning service to perform license plate recognition tasks on images. They prefer this route rather than teaching their staff to create and train advanced machine learning models properly and then incurring the cost of hosting a machine learning service to conduct this single task.

Contoso 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 customer bills. Any images containing license plates that the ready-made machine learning service cannot automatically detect 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 toll booth Id. Contoso has a customer service department that can monitor the queue of photos marked for manual validation and enter the license plates into a web-based form to be exported along with the automatically processed license plate data.

The process to export license plate data also needs to be automated. Contoso would like an automated workflow that runs on a scheduled interval to extract new license plate data received since the last export and save it into a CSV file that gets ingested by the billing software. They already have the CSV ingestion 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 provide the flexibility to increase or decrease the period as needed. This interval is based on the automated file ingestion process used by the billing system.

Customer service has requested that an alert email 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 provide them with 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, Contoso would like to have a centralized monitoring dashboard 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 to IT staff if anything goes wrong.

"Our directors want to explore 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."

As a stretch goal, Contoso would like to know that the license processing pipeline they have implemented is extensible to any number of future scenarios that are made possible once a license plate has been successfully processed. One scenario they have in mind is how the pipeline could support more advanced analytics. Contoso is looking for the capability to process data in a streaming fashion and process historical license plate capture events in a batch fashion (e.g., that could scale to analyze the historical data in the 10's of terabytes). They are curious if these analytic scenarios could also be implemented using a serverless architecture.

Customer needs

1. Replace its manual process with a reliable, automated solution using serverless components.

2. Take advantage of a machine learning service that would accurately detect license plate numbers without needing artificial intelligence expertise.

3. Provide a mechanism to manually enter license plate data from images that failed processing with the automated system.

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 develop the serverless components locally 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.

8. Design an extensible solution that could support serverless batch and real-time analytics, as well as other scenarios in the future.

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 can infinitely scale put us at risk for substantial monthly bills?

3. How do we make sure that erroneous image processing does not make specific toll bills fall through the cracks or, even worse, send an invoice to the wrong person?

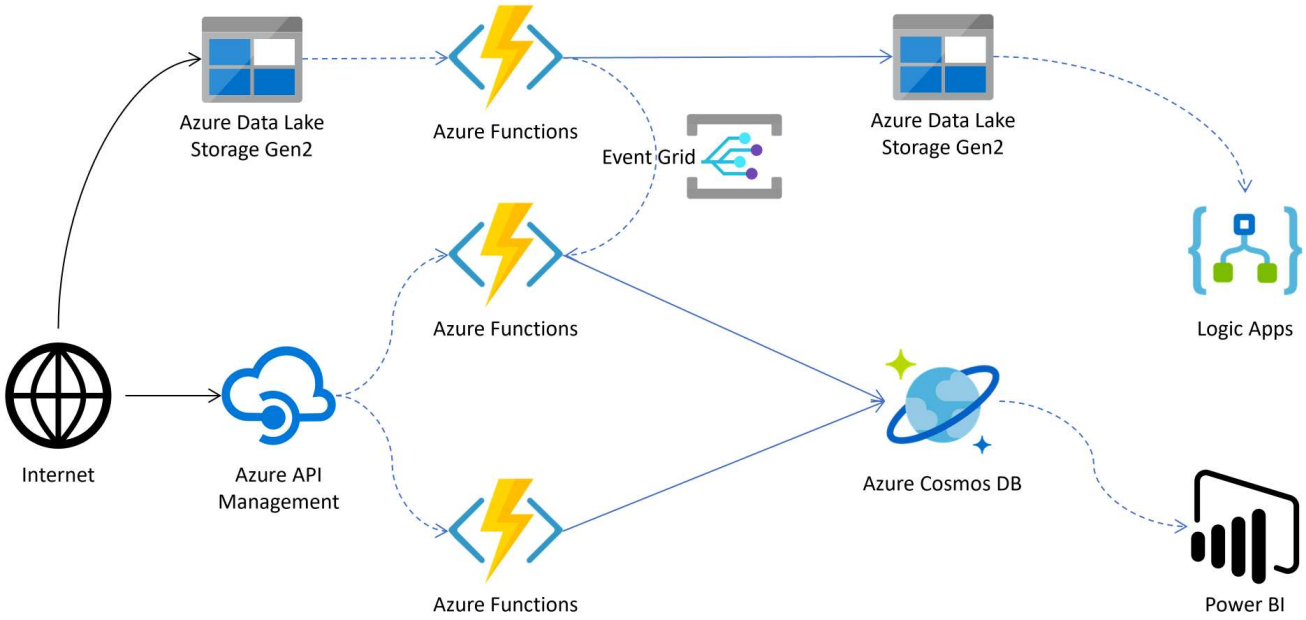
4. We are considering creating an API to allow our customers to retrieve information about their bills and photos that captured their vehicle but are concerned about unauthorized access and the possibility of a denial of service attack due to an excessive number of requests. Is there an Azure service that would enable us to implement a secure API that we can also monitor and manage, protecting our APIs from unauthorized access and excessive requests?

5. What is our best option to protect application secrets, such as connection strings, from being viewed by unauthorized users in the portal?

6. There appears to be some functional overlap between Azure Functions and Logic Apps. Can you help us to understand better when to use each and for what?

7. During our research on serverless architectures, we have read about the possibility of increased latency due to cold starts. Will this be an issue with Azure Functions?

Infographic for common scenarios



Step 2: Design a proof of concept solution

Outcome

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

Timeframe: 60 minutes

Business needs

Directions: With your team, answer the following questions and be prepared to present your solution to others:

1. Who will you present this solution to? Who is your target customer audience? Who are the decision makers?

2. What customer business needs do you need to address with your solution?

Design

Directions: With your team, respond to the following questions:

High-level architecture

1. Without getting into the details (the following sections will address these), 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 Contoso 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 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 Contoso use to conduct 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 into 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?

Extensible serverless analytics

1. Assuming they would like to be able to plug-in more solutions that respond to the event when a license plate has been successfully extracted from an image, how would you extend your solution using Event Grid? Be specific on the system topics, custom topics, and subscriptions at play.

2. What pipeline would you plug-into an Event Grid subscription listening for license plate events that could be used to provide real-time and batch analytics as a serverless solution?

Monitoring and DevOps

1. What tools and services would you recommend Contoso 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

Directions: As a team:

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.

Wrap-up

Timeframe: 15 minutes

Directions: Reconvene with the larger group to hear the facilitator/SME share the preferred solution for the case study.

Additional references

Description	**Links**
Introduction to Azure Functions	https://learn.microsoft.com/azure/azure-functions/functions-overview

What is Azure Logic Apps?	https://learn.microsoft.com/en-us/azure/logic-apps/logic-apps-overview
About Azure Cosmos DB	https://learn.microsoft.com/azure/cosmos-db/introduction
Choose between Azure services that deliver messages	https://learn.microsoft.com/azure/event-grid/compare-messaging-services
Monitor Azure Functions using Application Insights	https://learn.microsoft.com/azure/azure-functions/functions-monitoring
What is Azure Event Grid?	https://learn.microsoft.com/azure/event-grid/overview
Azure Event Grid bindings for Azure Functions	https://learn.microsoft.com/azure/azure-functions/functions-bindings-event-grid
Call Azure Functions from logic apps	https://learn.microsoft.com/azure/logic-apps/logic-apps-azure-functions
Azure Cosmos DB + Azure Functions	https://learn.microsoft.com/azure/cosmos-db/serverless-computing-database
Continuous deployment for Azure Functions	https://learn.microsoft.com/azure/azure-functions/functions-continuous-deployment
Code and test Azure Functions locally	https://learn.microsoft.com/en-us/azure/azure-functions/functions-develop-local
About Azure Key Vault	https://learn.microsoft.com/azure/key-vault/key-vault-overview
Use Key Vault references for Azure Functions	https://learn.microsoft.com/azure/app-service/app-service-key-vault-references
What is Azure AI Vision?	https://learn.microsoft.com/en-us/azure/ai-services/computer-vision/overview