Heartland Escapes Network Design

Aidan Polivka

CS-635: Computer Networking

Colorado Technical University

December 10, 2023

Table of Contents

Project Outline

“Heartland Escapes” is a bookstore in Nebraska. They've been in Lincoln for years, opening their first store in 2010 and their second in 2019. Their recent popularity is thanks to a fresh marketing approach on social media platforms like Instagram and Tik Tok. Events like the 'school’s out reading program,' author meet-and-greets, and Halloween story readings have contributed to their success. They're planning to expand to Omaha and Grand Island.

Since their 2009 opening, 'Heartland Escapes' has made minor technology upgrades, like increasing disk space, RAM, and upgrading the processor. Their current system, hosted at their first store, includes a Windows 2008 server with an Intel Core i7-8700 processor. This processor has 6 cores and 12 threads, running at 3.2 GHz, with an L2 Cache of 256 KB per core and an L3 Cache of 12 MB. The server has 8 GB of RAM and 8 TB of Hard Disk space, with 3.5 TB in use. It hosts a point-of-sale system, an inventory system, and a public website, all written in .NET Framework 4.8 web forms. It also includes a Microsoft SQL Server for the inventory and accounting databases. A network of 8 machines between the two locations uses the Point-of-Sale web application and Inventory API. The website interacts with the inventory API so users can search for books in each store from their homes. With two new stores and a growing customer base, 'Heartland Escapes' plans to upgrade their system for scalability and ease of management by moving to the cloud.

Given their Microsoft-centric system and growth expectations, I recommend migrating their infrastructure to Azure while keeping an on-premises server as a backup. This shift to the cloud supports hardware scalability for increased traffic without requiring dedicated CPUs. Azure, a Microsoft product, aligns with their .NET applications and MS SQL Server database management system.

There are two different types of scaling options for Azure SQL Server databases, DTU or vCore. For ‘Heartland Escapes’, we expect to use vCore for a couple reasons. The vCore pricing option is cheaper, and it follows in suit with the scaling procedures in the Azure App Services, keeping things simple across our

system. The General-Purpose tier offers 2-128 virtual cores, with 5.1 GB of RAM per virtual core. Our SQL Server will have a maximum of 10 virtual cores.

I suggest configuring Azure into two resource groups: one for data and infrastructure, and the other for applications. All resources should be in the us central region. In the application storage container, there will be three app services: one for the Point-of-Sale web application, one for the Inventory API, and one for the public website. Azure App Services are like containers with a base operating system. In this case, I recommend running all services on a Windows system for simplicity. The app services would use the Premium v3 P1V3 hardware tier, which includes two vCPUs and 8 GB of RAM per instance, with the ability to scale up to 30 instances. In the data and infrastructure group, there would be an Azure SQL Server hosting both databases, and a virtual network (vNet) that manages firewall rules and security. A vNet Gateway would allow users to access the system via VPN. The public website would be the only part of the system outside of the vNet, accessible only through the gateway. The public website's IP address would also have access.

Distributed Network Requirements Analysis

Migrating to a distributed environment fulfils a lot of 'Heartland Escapes' business requirements in their expansion effort. In this section, we'll break down five distinct business requirements and outline the ways in which migrating to Azure Platform as a Service offers solutions to them.

The first and most important requirement of 'Heartland Escapes' is *scaling and flexibility*. With 'Heartland Escapes' business model being centered around events; they go through periods of incredibly high customer volume. This includes traffic through their public website, and their critical inventory system. Azure offers the ability to scale horizontally, adding service nodes when hardware resources reach a configurable threshold. With our current service plan, we can configure their app services to scale out to up to 30 instances across all three of 'Heartland Escapes' services. This is significantly more computational power than their current on-premises system allows for and should serve them well. An added benefit to Azure as a service is that we can easily upgrade their plan if 'Heartland Escapes' grows at an unexpected rate.

The next business requirement that moving to a distributed system fulfils is *redundancy and high availability*. Azure App Services allows for deployed services to have geographic redundancy. This redundancy serves multiple purposes: geographic redundancy allows for users to access services from servers geographically near them, offering higher network speeds. Redundancy also allows for fault tolerance from natural disasters, so if something were to happen to one of Microsoft's server facilities, another facility will pick up the load from your business's deployed services. Microsoft's app service plans boast an uptime percentage of 99.95% (Ekuan), making 'Heartland Escapes' availability potentially higher than their on-premises system depending on their maintenance schedule.

Another business requirement for 'Heartland Escapes' is *Security and Access Control*. Azure offers Azure Active Directory, which will allow for fine-tuned control of user access and authentication. This will benefit 'Heartland Escapes' greatly in controlling which individuals can have database read and/or write

access, access to the Point-of-Sale system, and granting accesses from services to connect to the Inventory API. Azure also allows lots of network security options like vNets and network gateways. This will allow 'Heartland Escapes' to control what service ports are open to be communicated with, and what IPs are able to communicate with resources within the virtual network. The virtual gateway will also allow employees using the Point-of-Sale system to connect to the virtual network via VPN and sign in securely.

The fourth business requirement needed when migrating to a distributed environment is *Resource Sharing and Centralized Data Management*. 'Heartland Escapes' shares a single SQL Server instance, and two distinct databases. This is central to their inventory management system and Point-Of-Sale system. This should not be a significant change to their architecture, however the SQL Server instance will no longer be hosted on the same machine as the services. Because these services are no longer hosted on the same machine, measure will need to be taken to ensure that only services within the virtual network can access the Azure hosted SQL Server.

The fifth and final business requirement for 'Heartland Escapes' migration effort is *Remote Access and Mobility*. Users of the Point-Of-Sale system will need to be able to access that service from within the Virtual Network, so a vNet Gateway will be required connect via VPN. 'Heartland Escapes' also hopes to be able to open mobile pop-up stores, so being able to connect from locations outside of the physical store will be a necessity. Their network structure will need consideration to make this functionality secure.

Security is of utmost importance for 'Heartland Escapes' since they are transferring financial data from their customers to third party payment processors. Measures will need to be taken to ensure that data in transit and at rest is encrypted per the Payment Card Industry Data Security Standard (PCI DSS). If compliance to this standard is not met, 'Heartland Escapes' risks large fines that are beyond the business's capability to pay. (Baykara, 2023)

Communication Protocols Analysis and Recommendations

We will want to make our choices of communication protocols with 'Heartland Escapes' business requirements in mind. Understanding that 'Heartland Escapes' transmits and stores customer financial information and potentially other personally identifiable information (PII), we'll want to ensure that the messaging protocols we subscribe to are highly secure. We also want to make sure that our protocol stack is robust and is not subject to change any time soon since this is a migration/modernization effort. Additionally, we'll need to consider what protocols Azure can communicate with. If we weren't migrating to a platform that's as established as Azure, we might have wanted to make this consideration prior to choosing Azure as a cloud host.

HTTPS and SSL/TLS Certificates

Because 'Heartland Escapes' system is primarily web based, we'll plan to use HTTPS for their web application protocols. HTTPS stands for "Hypertext Transfer Protocol Secure" and is an extension of its predecessor HTTP. HTTP was used for a long time but was considered insecure because it doesn't use any form of encryption of hypertext messages in transit. This security issue has been resolved by the HTTPS extension of HTTP, which encrypts the hypertext message to a cyphertext using SSL/TLS certificates. In fact, this security improvement to the original HTTP protocol has just about replaced HTTP entirely, and most modern web browsers will flag web pages as "not secure" if you try to access a domain that uses HTTP. (GeeksForGeeks, 2022)

So, what is SSL/TLS, and what does it add to the HTTP protocol to make it secure? SSL is an acronym for "Secure Socket Layer". This layer is an encryption-based internet protocol layer, used to secure messages across the web. The layer uses an authentication method called a "handshake" between the sending and receiving service before sending the message data. This handshake ensures that the two machines are who they say they are, rather than a bad actor claiming to be an authenticated machine in the network. As an

added layer of security, the SSL protocol digitally signs all messages to the receiving machine to ensure that the message wasn't tampered with in transit. (CloudFlare) As with any technology, the SSL protocol has undergone updates. The updated version of SSL is TLS, which stands for "Transport Layer Security". In the networking world those names have become synonymous, hence the name SSL/TLS. For example, the website DigiCert sells all their TLS certificates under the name SSL, although it is really a TLS certificate you're receiving. (digicert)

HTTPS and SSL/TLS will be used all over the new 'Heartland Escapes' system. It will be used in communication between app services, between app services and client web browsers, and between app services and the Azure hosted database. This is to ensure security of the message data sent, and to make sure that the services communicating amongst each other are who they say they are.

OpenVPN Protocol

Many VPN providers utilize a protocol called "OpenVPN". This protocol uses a custom protocol based on SSL/TLS to secure communication between machines. It also allows for two-factor authentication and a kill switch to ensure that individuals connecting are authenticated. OpenVPN is an open-source protocol, which means that its code base is available to be read by the public. This has its pros and cons, the benefit being that consumers can access the code and ensure that it's up to their standards, the cons being that bad actors can read the code and try to find ways to break it. OpenVPN allows for connection from most devices, including windows, linux, and mac computers, and android and apple phones. OpenVPN can access email, file servers, and databases. (GeeksForGeeks, 2023)

OpenVPN is one of the protocol options in Azure's Point-To-Site VPN, and it's clearly the best options for 'Heartland Escapes'. With it being one of the most secure VPN protocols available, and capable of accessing so many internal resources, it would be the perfect solution for 'Heartland Escapes' future

endeavors to allow pop up stores in temporary locations. It also helps ‘Heartland Escapes’ adhere to the requirements of the Payment Card Industry Data Security Standard.

Network Traffic Analysis and Recommendations

'Heartland Escapes' system is completely distributed, meaning that every service at some level requires a network traffic analysis. For this section, we'll discuss each service, the use cases of the service, its network traffic requirements, and potential congestion points. 'Heartland Escapes' is comprised of four major services, the public website, the inventory API, the point-of-sale system, and the Azure SQL server. Before getting into the specifics of each service, we must preface this analysis with our architecture, and how Azure as a host affects what mitigation methods are available for 'Heartland Escapes'.

Congestion Mitigation and Azure Platform as a Service

Because this enterprise's system is hosted by Azure as a cloud provider, we have limited options when considering congestion mitigation methods. In a traditional distributed network with static hardware resources, it would make sense to attempt to mitigate congestion through protocol options and traffic management methods like Random Early Detection or Explicit Congestion Notifications. 'Heartland Escapes' happens to be in a position where the hardware resources are highly scalable, and able to be scaled on an automated basis under CPU, memory usage and time of day conditions. We'll plan on taking advantage of Azure App Service scaling options for anticipated high traffic times.

Azure offers additional methods to manage network traffic, and Azure App Services have some integrated traffic management methods. Azure Traffic Manager is used to distribute traffic to public applications across all Azure regions (Lindsay, 2023). Because 'Heartland Escapes' system is limited to Eastern Nebraska, Azure tools like Azure Traffic Manager aren't needed. Additionally, Azure App Services provide round-robin functionality for nodes within the same region. Even when scaled down to a single instance, the other available nodes continue to serve requests (Lin, 2020).

Considering the App Service Plan that all our services are operating under, 'Heartland Escapes' system has up to 30 total nodes available between the three deployed applications. The base configuration

for each of our applications will be a minimum of two nodes, scaling up at 80% CPU or 80% memory utilization, and scaling down at 40% CPU or 40% memory utilization. This leaves the system with 24 additional nodes for peak traffic times and hardware required scaling.

Public Website and Inventory API

The public website is the only service that is open for any client to use. This site serves multiple purposes, including finding store hours and locations, discovering event schedules for each store, and exploring inventory across all store locations. The inventory searching capability has a high level of impact on the Inventory API because the API is the data source for inventory information. The public website does not communicate directly with any databases to keep accessibility to the SQL Server strictly within the virtual network.

High traffic times differ between weekdays and weekends. Because 'Heartland Escapes' is geographically located in eastern Nebraska, network traffic spikes are limited to the Central Daylight time zone. Weekdays spike between 11:00 am and 1:00 pm, and between 4:00 pm and 7:00 pm. Weekend traffic spikes between 10:00 am and 3:00 pm, which is consistent with 'Heartland Escapes' in store foot traffic. There is a consistent amount of traffic outside of spike times between 8:00am and 9:00pm Saturday through Sunday, with the night traffic being relatively non-existent. During peak times, the public website and the Inventory API should scale up to a minimum of four service nodes.

Point-of-Sale

The point-of-sale system is used exclusively by employees of 'Heartland Escapes'. This system's purpose is specifically to charge customer cards, make updates to the inventory database via the Inventory API, and to store data in the Accounting Database. This system is currently used only in store but is expected to be used in pop-up stores as 'Heartland Escapes' grows.

The point-of-sale system experiences the same traffic peaks as the public website, but not the same amount of volume. Because the point-of-sale is limited to the four registers at each store, there is a maximum of 16 machines using the system at a time, which shouldn't cause a need for time-based scaling configurations. The only application that might suffer from this would be the Inventory API, which will be affected by both the point-of-sale system and the public website at the same time. I don't believe that the impact from the point-of-sale system would be substantial enough to warrant increasing the minimum node amount. If it reaches the point where the Inventory API is at increased resources, our default rules for max CPU and memory will go into effect.

Azure Hosted Microsoft SQL Server

The SQL Server is the true data source for all Inventory data. So, the peak times for all services interacting with the Inventory API dictates the peak times for the Inventory Database on our Azure SQL Server. There is an additional database that's interacted with solely by the point-of-sale system. This accounting database's purpose is to log financial transactions. We're yet again in the situation where physical scaling out of resources will be the primary method of resolving any form of network congestion because of our cloud host.

Because we're using serverless computation for our database architecture, the compute resources will scale as needed. We're yet again hindered by some of the black-box functionality of azure as to how it manages routing to different computational nodes in our database model. If a need arises to have read-only scaling of our Inventory database, we could upgrade our SQL Server to the Business Critical service tier, but I don't anticipate network traffic to be our limiting factor for our SQL databases (Setlem, 2023).

Final Recommendation

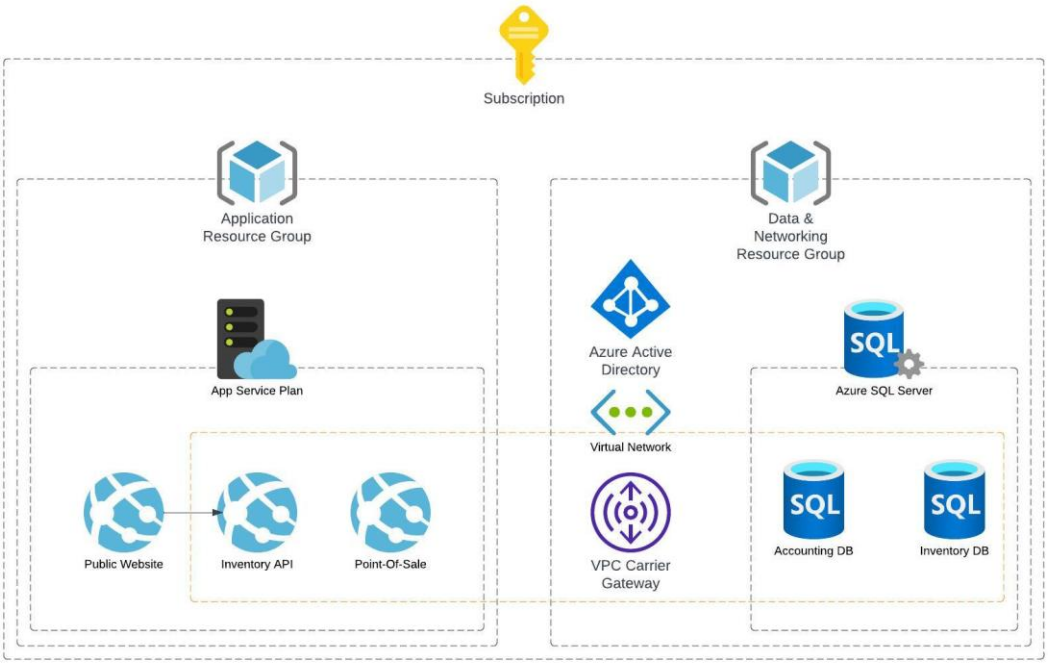
My advisement for 'Heartland Escapes' network traffic management unfortunately (or fortunately depending on your perspective) is to manage it completely by hardware scaling. This is the recommendation

by Azure (Sangapu, 2023), and therefore my recommendation for ‘Heartland Escapes’. Each service should have a minimum of two service nodes, and the Inventory API and the public website should scale up to four service nodes Monday through Friday between 11:00 am and 1:00 pm and between 4:00 pm and 7:00 pm, and Saturday and Sunday between 10:00 am and 3:00 pm. These additional nodes should allow for more routing availability and throughput.

Network Design and Architecture

Because ‘Heartland Escapes’ new system is distributed in Azure, all our cost concerns can be easily calculated using the Azure Price Calculator. In this section, we’ll outline the Azure architecture that will be the host of our system and a somewhat general topography of the system. This topography will follow client requests through both the Point-Of-Sale system, and the public website. We’ll also run a cost analysis of the Azure resources used in this new distributed system, and what the month-to-month cost of operation of the enterprise will be.

Starting with our Azure Resource Architecture, this is a general distribution of each Azure component within their Azure subscription. This will help us with our cost analysis of the Azure system in that we’ll have insight into all the components that involve monthly costs. It will be difficult to know exactly how much this system will cost a month because of the Pay-As-You-Go nature of Azure and horizontal scaling app service scaling. Generally, the system will use two instances per app service, so that will be the assumption for our cost analysis.



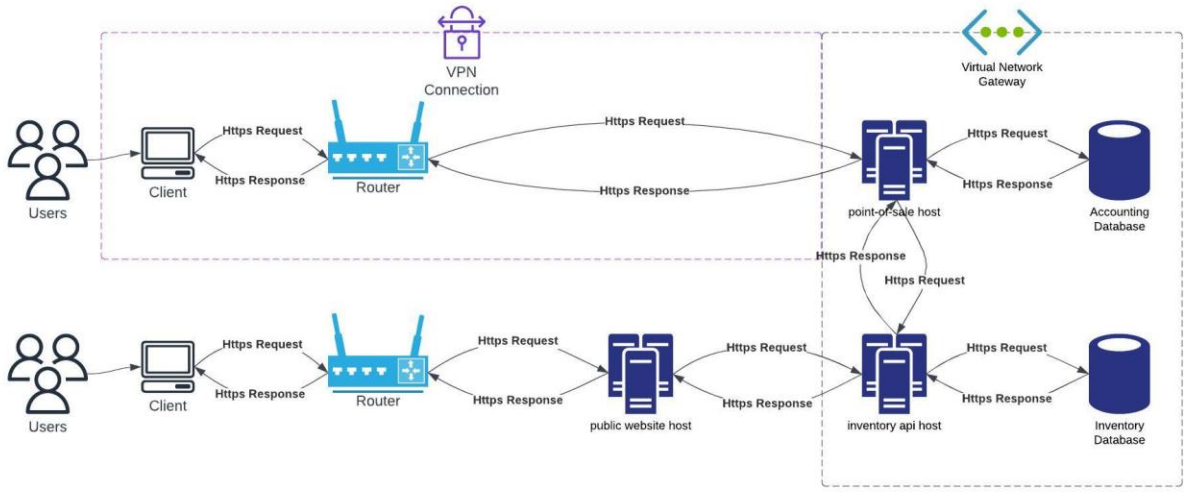
We have the resource groups split between the application resources, and the data/infrastructure resources for a couple reasons. For cost, it's easier to break down the application cost versus the infrastructure and data. Also, it segments the cloud managers work so that it's organized, and there is separation of concerns in terms of cloud administrative work. It's important to note, these resource groups have no bearing upon security, implementation, or infrastructure. These are simply for the Azure cloud administrator's ease of operation, and potentially cost querying capability. Within the diagram is a hint toward what virtual network access would look like, and what services will reside within the virtual network.

So, what is the cost breakdown? Here we'll outline what each service's cost is monthly, and what the overall cost for 'Heartland Escapes' will be on a month-to-month basis. This is not a breakdown of the cost of the migration effort, which will require the work of multiple engineers, database administrators, and cloud administrators. At the base operation cost, this is what Heartland Escapes is looking at from a month-to-month basis assuming that the default hardware resource configurations are sufficient regardless of peak activity.

Service category	Service type	Custom name	Region	Description	Estimated monthly cost
Identity	Azure Active Directory External Identities	Heartland Escapes AD	West US	Premium P1 tier: 120 monthly active user(s), 0 SMS/Phone Events	$0.00
Networking	Virtual Network	Internal Vnet		Central US (Virtual Network 1): 100 GB Outbound Data Transfer; South Central US (Virtual Network 2): 100 GB Outbound Data Transfer	$14.00
Networking	VPN Gateway	VPN	Central US	VPN Gateways, Basic VPN tier, 750 gateway hours, 10 S2S tunnels, 128 P2S connections, 500 GB, Inter-VNET outbound VPN gateway type	$44.50
Databases	Azure SQL Database	Accounting & Inventory Databases	Central US	Single Database, vCore, General Purpose, Serverless, Standard-series (Gen 5), Locally Redundant, 4 Billed vCores, RA-GRS Backup Storage Redundancy, 0 GB Point-In-Time Restore, 0 x 5 GB Long Term Retention	$9.57
Compute	App Service	Inventory API	Central US	Premium V3 Tier; 2 P1V3 (2 Core(s), 8 GB RAM, 250 GB Storage); 3 year savings plan; Windows OS; 3 SNI SSL Connections; 0 IP SSL Connections; 0 Custom Domains; 0 Standard SLL Certificates; 0 Wildcard SSL Certificates	$370.12
Compute	App Service	Public Website	Central US	Premium V3 Tier; 2 P1V3 (2 Core(s), 8 GB RAM, 250 GB Storage); 3 year savings plan; Windows OS; 0 SNI SSL Connections; 0 IP SSL Connections; 0 Custom Domains; 0 Standard SLL Certificates; 0 Wildcard SSL Certificates	$370.12
Compute	App Service	Point-Of-Sale	Central US	Premium V3 Tier; 2 P1V3 (2 Core(s), 8 GB RAM, 250 GB Storage); 3 year savings plan; Windows OS; 0 SNI SSL Connections; 0 IP SSL Connections; 0 Custom Domains; 0 Standard SLL Certificates; 0 Wildcard SSL Certificates	$370.12
Support			Support		$0.00
			Licensing Program	Microsoft Customer Agreement (MCA)	
			Total		$1,178.44

As you can see, this is going to be a pretty low cost solution for ‘Heartland Escapes’ month-to-month, especially considering the ease of scalability if they continue to grow at the rate they anticipate. The number of app service instances and the computational power of each of those instances is probably more than will be needed for now. The cost calculation for the VPN minutes and vNet data transfer were aggressive, I don’t believe that ‘Heartland Escapes’ would need 500 GB of vNet outbound data transfer bandwidth. ‘Heartland Escapes’ is also saving a lot of money using serverless computation for the SQL server, rather than having provisioned resources. The cost difference is well over $1000 dollars a month, where now their database cost is going to be less than a Netflix subscription.

In a deployed scenario with all these resources created and operating, this would be the flow of the ‘Heartland Escapes’ network. Some configurations will be needed to ensure the service for the public website has access into the vNet, and that authentication into the VPN is secure. This is a general network topography for the ‘Heartland Escapes’ system, the first client being the Point-of-Sale system, and the second client being the public website.



Future Needs Analysis and Recommendations

TBD

References

Baykara, S. (2023, October 9). PCI DSS control objectives. PCI DSS GUIDE. https://pcidssguide.com/pci-dss-control-objectives/

Ekuan, M. (n.d.). Azure App Service and reliability. Microsoft Azure Well-Architected Framework | Microsoft Learn. https://learn.microsoft.com/en-us/azure/well-architected/service-guides/azure-app-service/reliability

GeeksforGeeks. (2022, May 6). HTTPS full form. GeeksforGeeks. https://www.geeksforgeeks.org/https-full-form/

GeeksforGeeks. (2023, January 24). Types of virtual private network (VPN) and its protocols. GeeksforGeeks. https://www.geeksforgeeks.org/types-of-virtual-private-network-vpn-and-its-protocols/

How does SSL work? | SSL certificates and TLS | cloudflare. CloudFlare. (n.d.). https://www.cloudflare.com/learning/ssl/how-does-ssl-work/

What is an SSL certificate?. DigiCert. (n.d.). https://www.digicert.com/what-is-an-ssl-certificate

Assaf, W. (2023, November 7). VCore purchasing model - azure SQL database. Microsoft Learn. https://learn.microsoft.com/en-us/azure/azure-sql/database/service-tiers-sql-database-vcore?view=azuresql

Lin, C. (2020, March 30). Control traffic with traffic manager - azure app service. Control traffic with Traffic Manager - Azure App Service | Microsoft Learn. https://learn.microsoft.com/en-us/azure/app-service/web-sites-traffic-manager

Lindsay, G. (2023, August 15). Azure traffic manager. Azure Traffic Manager | Microsoft Learn. https://learn.microsoft.com/en-us/azure/traffic-manager/traffic-manager-overview

Sangapu, M. (2023, January 25). Troubleshoot performance degradation - azure app service. Troubleshoot performance degradation - Azure App Service | Microsoft Learn. https://learn.microsoft.com/en-us/azure/app-service/troubleshoot-performance-degradation

Setlem, R. (2023, November 28). Read queries on Replicas - Azure SQL Database & SQL Managed instance. Read queries on replicas - Azure SQL Database & SQL Managed Instance | Microsoft Learn. https://learn.microsoft.com/en-us/azure/azure-sql/database/read-scale-out?view=azuresql

Microsoft. (2023). Pricing calculator: Microsoft Azure. Pricing Calculator | Microsoft Azure. https://azure.microsoft.com/en-us/pricing/calculator/?ef_id=_k_7241ec353e0e15dcfcdf89434f450a2f_k_&OCID=AIDcmm5edswduu_SEM__k_7241ec353e0e15dcfcdf89434f450a2f_k_&msclkid=7241ec353e0e15dcfcdf89434f450a2f