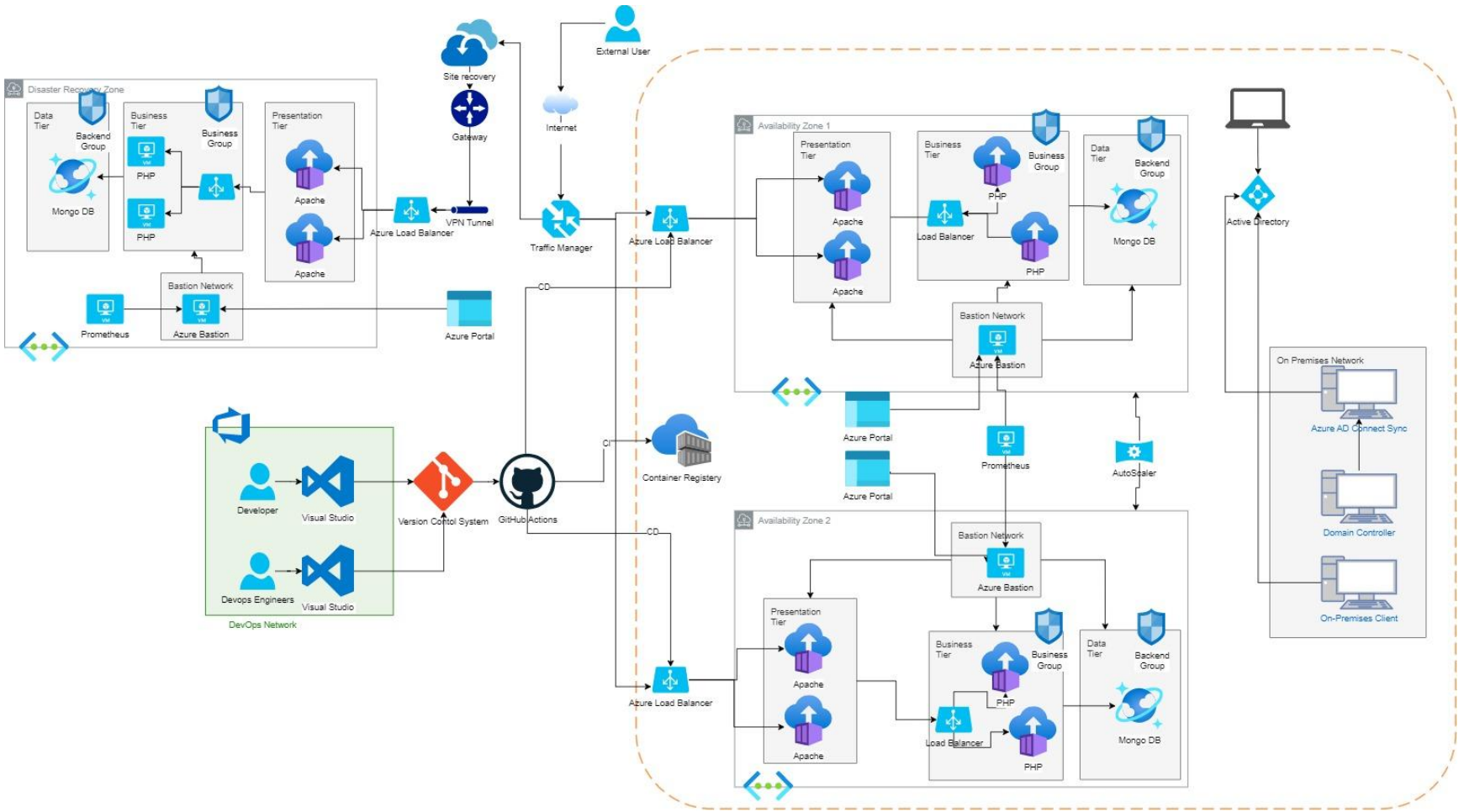
Cloud Design:

A) Platform Design:

In this cloud-based architecture, The proposed solution for Runners Crisps' promotional campaign begins with external users accessing the website through a traffic manager. This traffic manager is responsible for directing users to one of the three availability zones, which include two availability zones and a disaster recovery zone, as well as an on-premises network. This design ensures optimal performance and fault tolerance during the European Cup final when traffic is expected to surge. Both availability zones share a similar architecture, starting with a load balancer connected to the traffic manager. This load balancer evenly distributes incoming traffic across two containerized presentation tier instances. These instances are responsible for handling user input, such as the 10-digit hexadecimal code and other required user information. Next, the presentation tier instances connect to another load balancer, which manages traffic to two container instances in the business tier. The business tier containers interact with the data timer using a web service that validates the user's 10-digit code and processes the submitted data. For the data tier, an azure cosmos database service is used to host the mongo database since it is a great choice for NoSQL databases. This ensures redundancy and scalability, as well as the efficient storage and retrieval of user information. Both the business and data tiers are secured by a network security group (NSG), and all tiers in each zone connect to a Bastion network for remote access and management. The Bastion networks in both availability zones are equipped with Prometheus for monitoring and are interconnected for seamless data exchange. Additionally, each Azure Bastion instance connects to an Azure portal for centralized administration. To handle fluctuations in traffic and maintain optimal performance, an autoscaler is integrated between the two availability zones. By employing this architecture, Runners Crisps can ensure a reliable and low-latency experience for users during the promotion, as well as effectively manage the application and access stored data for post-processing. In order to manage user identities and access, the Traffic Manager is connected closely with Azure Active Directory (Azure AD). Azure AD is integrated with Azure AD Connect Sync, Domain Controller, and on-premises clients to ensure a smooth and efficient user authentication and authorization process throughout the whole infrastructure. To ensure business continuity and protect against disaster scenarios, the Traffic Manager is also connected to an Azure Site Recovery service. The Site Recovery service is linked to a VPN tunnel, which connects to a standby Availability Zone. This zone mirrors the other Availability Zones' configurations but does not have the auto-scaling mechanisms in place. In the event of a failure or outage, the Site Recovery service can quickly failover to the standby zone, minimizing downtime and ensuring that the system remains operational.



B) Platform Evaluation:

	Azure	Google Cloud

Gateway/Load Balancer	Azure Gateway & Azure Load Balancer	Cloud Load Balancing
	- Better integration with Azure services - Suitable for organizations using Azure	- Works seamlessly with Google Cloud services
VPN Connectivity	Azure VPN Gateway	Cloud VPN
	- Secure, encrypted communication - Seamless integration with Azure services	- Secure, encrypted communication
Container Management	Azure Container Instances	Cloud Run
	- Rapid deployment and simplified container management - Native integration with Azure Load Balancer and Azure Gateway	- Fully managed container service
Integration & Ease of Use	- Easy integration and compatibility with other Azure services - Consistent and cohesive experience for organizations using Azure	- Easy integration and compatibility with other Google Cloud services
Market Presence	- Long-standing presence in the market - Extensive range of services	- Innovative solutions in specific areas, such as machine learning and artificial intelligence
AI and Data Science	-Strong in AI,ML,and Analytics Services -Azure ML and Azure Synapse Analytics - Azure AI and Cognitive Services for AI development	-Strong in AI,ML,and Analytics Services - Tensorflow and Bigquery - Platform for end-to-end AI development

Vendor Lock-in	- Seamless integration may lead to higher lock-in risk - Easier to migrate to and from other platforms	- Less likely to cause lock-in due to its more open approach - Migration to other platforms may be more challenging
Anti-Competitive practices	- A smaller market share compared to Azure - More likely to engage in anti-competitive practices	- Larger market share compared to Google Cloud - Less likely to engage in anti-competitive practices
Overall Advantage	- Better for organizations already invested in the Azure ecosystem, offering a comprehensive cloud solution	- Equally capable but more focused on specific areas of innovation

The advantages of using Azure tools over their Google Cloud counterparts can be attributed to seamless integration and compatibility with other Azure services, as well as Azure's long-standing presence in the market and its extensive range of offerings. This makes Azure an attractive option for organizations requiring a comprehensive cloud solution. While Azure's tightly knit ecosystem can make it easier to integrate applications, it might also make it more challenging, should the need arise, to switch to another platform like Google Cloud. This reliance on a single source could make it more difficult for an organization to benefit from new developments and upcoming technologies made accessible by other platforms. Large cloud providers like Azure and Google Cloud may engage in anti-competitive behavior by putting barriers in the way of smaller rivals. This could potentially hinder innovation and lessen the range of cloud solutions that are offered to consumers. Even though Google Cloud tools are equally capable and offer innovative solutions in specific areas such as machine learning and artificial intelligence, Azure tools might be more appealing to beginners due to their consistency and a reduced learning curve associated with adopting a new cloud platform such as AI and Machine Learning.

C) Architectural options:

Services (Faas,Paas,etc):

The system is a combination between Faas and Paas. FaaS services, such as Azure Container Instances, provide rapid deployment and simplified container management, reducing operational overhead and infrastructure management. PaaS services, such as Azure

Traffic Manager and Load Balancer, abstract away underlying infrastructure and management complexities. A clear overview of what these services does can be found below:

Starting with Azure Gateway, it is used to provide a public IP address and act as the entry point for user traffic. It offers secure communication between the users and the backend services. The Azure Load Balancer is utilized to distribute incoming traffic evenly across frontend containers. The utilization of these resources is optimized and application responsiveness is improved through the use of Azure Load Balancer. This tool benefits from being natively integrated with other Azure services, including Azure Gateway and Azure VPN, resulting in a seamless and unified experience. The secure connections between zones are established using the Azure VPN Gateway, which establishes an encrypted communication channel that effectively prevents unauthorized access and data breaches. Azure Container Instances are employed to host the frontend and backend components of the system. This serverless container service enables rapid deployment and simplifies container management. Google Cloud offers Cloud Run as an alternative, which is also a fully managed container service.

Microservices:

The solution is designed around a microservices pattern, with separate front-end and backend containerized services. This approach provides several benefits, including better resource utilization, increased scalability, and independent deployment and management of individual services. Nevertheless, managing microservices may introduce additional complexity and coordination overhead compared to a monolithic architecture.

Methods of Access:

User access is managed through the Traffic Manager, which serves as the primary entry point and distributes incoming requests across Availability Zones. This approach improves fault tolerance and responsiveness. However, it may also introduce additional latency and complexity compared to a single entry point.

Security Considerations:

The architecture employs diverse security measures, including Azure VPN Gateway for secure connectivity, Azure Active Directory for identity and access management, and Azure Site Recovery for disaster recovery. While these services provide strong security, they may also pose potential vulnerabilities and necessitate ongoing monitoring and upkeep to guarantee compliance with industry standards. Furthermore, the utilization of NSGs for the business and data tiers as well as Bastion networks for remote access and

management helps ensure secure system access. Additionally, the integration of Prometheus enables system performance monitoring and detection of possible security concerns.

Load Balancers and Auto-Scaling:

Load Balancers and auto-scaling mechanisms were chosen to ensure optimal resource utilization and application responsiveness. These design decisions contribute to the architecture's overall resilience and scalability. Load balancers distribute incoming traffic evenly across multiple instances within each tier, preventing overloading of individual instances and ensuring optimal performance. In the proposed design, load balancers are employed at the presentation, business, and data tiers. By incorporating load balancers at each tier, the architecture is able to handle fluctuating traffic patterns effectively, providing a consistent and low-latency user experience. Auto-scaling mechanisms dynamically adjust the number of instances within each tier based on real-time demand. In the proposed design, an auto-scaler is integrated between the two availability zones. By automatically scaling instances up or down, the architecture can efficiently allocate resources, thereby preventing both under-utilization and over-provisioning. This capability allows the system to accommodate traffic surges during the European Cup final while minimizing operational costs.The use of Azure cosmos db also provides a high autoscaling mechanism. Azure Cosmos DB is a globally distributed fully managed NoSQL database service which is going to be suitable for the cloud solution proposed since the database will be using mongo db as a NoSQL database. It is designed to provide low latency, high availability, and seamless horizontal scaling across multiple regions, ensuring consistent performance for your applications. Cosmos DB is ideal for various application types, from web and mobile apps to IoT and gaming solutions.

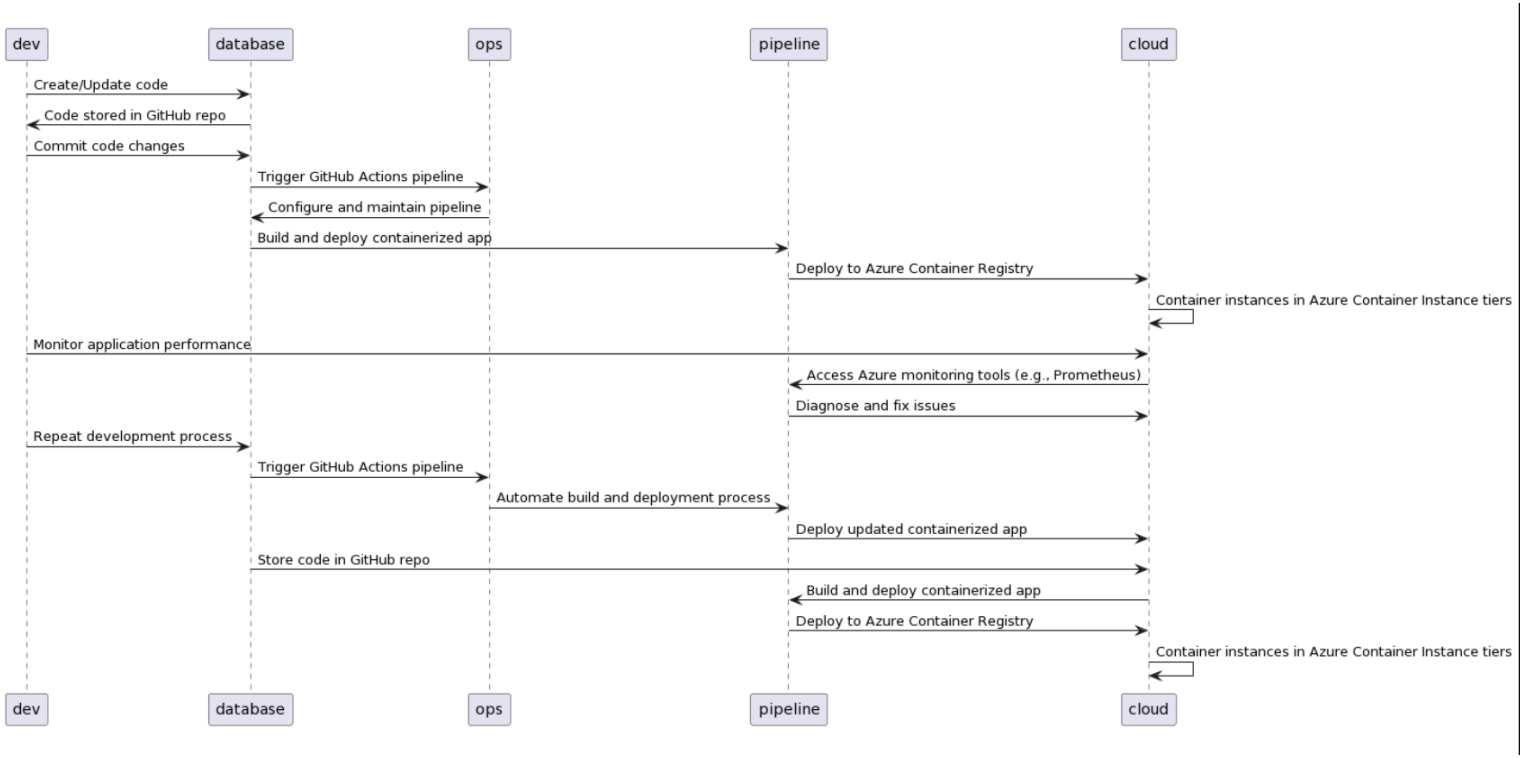
Monitoring and Observability:

In the proposed design for Runners Crisps' campaign, monitoring and alerting methods play a crucial role in ensuring system reliability, performance, and security. The primary monitoring and alerting tool employed in the architecture is Prometheus. Prometheus is integrated into the Bastion networks of both availability zones, enabling comprehensive monitoring of the system components. As a powerful open-source monitoring and alerting toolkit, Prometheus collects valuable metrics and insights into the performance and health of the presentation, business, and data tiers. Its integration with the Bastion networks allows for seamless data exchange between the zones, providing administrators with a holistic view of the entire system's operation. By leveraging Prometheus, administrators can effectively detect and address potential issues before they escalate, helping to maintain the system's reliability and low-latency user experience during the high-traffic European Cup final. In addition to Prometheus, the design also utilizes Azure portal connections for centralized administration. Through the Azure portal, administrators can access monitoring data, logs, and alerts generated by the various Azure-native services, such as Azure Load Balancer, Azure Container Registery, and Azure Container Instances. This centralized access facilitates a more efficient management of the system's performance and security.

Cloud Application:

A) Source control Automation:

The GitHub Actions pipeline serves an important function in constructing and deploying the containerized application to Azure container instances. DevOps engineers assume responsibility for the configuration and maintenance of the GitHub Actions pipeline, guaranteeing its proper configuration for the build and deployment of the application. Consequently, the diagram demonstrates a connection between the DevOps user and the GitHub Actions pipeline, highlighting their role in ensuring the operation of the pipeline. Developers bear the responsibility of creating and updating the application code, which is stored in the GitHub repository. Utilizing their development environments, developers write code on visual studio or whatever IDE they are using, test the software locally, and then commit the changes they have made to the repository. Upon completion of updates and readiness for deployment, developers commit the alterations to the GitHub repository. Subsequently, the GitHub Actions pipeline initiates an automated process that builds and deploys the containerized application to the Azure Container Registry and the corresponding Azure Container Instance tiers. Post-deployment, developers may be required to monitor the application's performance and address any emergent issues. To achieve this, they can access the container instances and employ Azure monitoring tools like the prometheus to monitor the application's status, examine logs, and diagnose potential problems. If additional updates to the application are deemed necessary, developers can recommence the development process and commit the changes to the GitHub repository. The GitHub Actions pipeline will subsequently automate the build and deployment of the updated application to the container registry and then the container instances, allowing the process to iterate as many times as needed.The diagram below shows the flow described in the text:



B) Project Evaluation:

Developing the cloud proof-of-concept for this architecture presented a variety of challenges, which were addressed using several strategies and technologies. One of the main challenges faced in project development is ensuring consistent performance and availability across multiple zones is a primary concern in this architecture. To overcome this challenge, two availability zones and a disaster recovery zone were made to allow high availability for users as well as using a traffic manager between zones and load balancers between containers. Accommodating high workloads was also another crucial challenge for maintaining system responsiveness and availability, particularly during periods of high demand such as the cup final. To address these issues, all containers have been applied to an auto scaling mechanism with load balancers and a traffic manager to distribute traffic across zones and containers, and providing a mechanism to handle crashes of the containers and helping site recovery using Azure Site

Recovery and VPN tunnels. Another problem was proactively identifying and addressing potential issues was essential for maintaining the stability of the distributed system. A Prometheus was used to overcome this challenge, as written in the architectural options, Prometheus enables monitoring cloud-native applications, microservices, and containerized environments. Furthermore, the architecture involves numerous components working together, including Azure Gateway, load balancers, frontend and backend containers, and the Prometheus-based monitoring solution. Smooth communication between these elements was a challenge that required careful planning, configuration, and testing.

References:

- Microsoft. (n.d.). Disaster recovery for enterprise-scale applications. Azure Architecture Center. Retrieved from https://learn.microsoft.com/en-us/azure/architecture/solution-ideas/articles/disaster-recovery-enterprise-scale-dr
- Microsoft. (n.d.). Site Recovery overview. Azure Site Recovery documentation. Retrieved from https://learn.microsoft.com/en-us/azure/site-recovery/site-recovery-overview
- Microsoft. (n.d.). What is Azure AD Connect? Azure Active Directory documentation. Retrieved from https://learn.microsoft.com/en-us/azure/active-directory/hybrid/whatis-azure-ad-connect
- Microsoft. (n.d.). Cross-Cloud Scaling. Azure Architecture Center. Retrieved March 28, 2023, from https://learn.microsoft.com/en-gb/azure/architecture/solution-ideas/articles/cross-cloud-scaling.
- Microsoft. (n.d.). Virtual machine scale sets - Autoscale overview. Microsoft Azure. Retrieved from https://learn.microsoft.com/en-us/azure/virtual-machine-scale-sets/virtual-machine-scale-sets-autoscale-overview
- Microsoft. (n.d.). GCP Professional Services. Retrieved from https://learn.microsoft.com/en-us/azure/architecture/gcp-professional/services
- Microsoft Corporation. (2021). Secure VNet DMZ reference architecture. Microsoft Azure Architecture Center. https://learn.microsoft.com/en-us/azure/architecture/reference-architectures/dmz/secure-vnet-dmz?tabs=portal
- Microsoft. (2021). Azure DevOps with AKS. Microsoft Learn. https://learn.microsoft.com/en-us/azure/architecture/example-scenario/apps/devops-with-aks