Logo, company name

Description automatically generatedSolution Architecture

Version 1.0

*Web site portfolio*

**Drone Shuttles L.L.C**

*February 2022*

# Document History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version No.** | **Date** | **Author** | **Revision Description** |
| 1.0 | 01-02-2022 | Hany Saad | Draft |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Contents**

[Document History ii](#_Toc94690060)

[1. Solution Overview 1](#_Toc94690061)

[1.1 As-Is Architecture 1](#_Toc94690062)

[1.2 To-Be Architecture 1](#_Toc94690063)

[1.3 Transition Planning 2](#_Toc94690064)

[2. Architecture Goals and Constraints 2](#_Toc94690065)

[2.1 Architectural Assumptions and Decisions 2](#_Toc94690066)

[2.2 Solution Architecture Attributes 2](#_Toc94690067)

[2.2.1 Technology 3](#_Toc94690068)

[2.2.2 Portability 3](#_Toc94690069)

[2.2.3 Capacity 3](#_Toc94690070)

[2.2.4 Performance 3](#_Toc94690071)

[2.2.5 Availability and Reliability 3](#_Toc94690072)

[2.2.6 Scalability 3](#_Toc94690073)

[2.2.7 System Management, Monitoring, and Administration 4](#_Toc94690074)

[2.2.8 BC/DR and COOP 4](#_Toc94690075)

[2.2.9 Security 4](#_Toc94690076)

[2.2.10 Cost Optimization 4](#_Toc94690077)

[2.3 Deployment 4](#_Toc94690078)

# Solution Overview

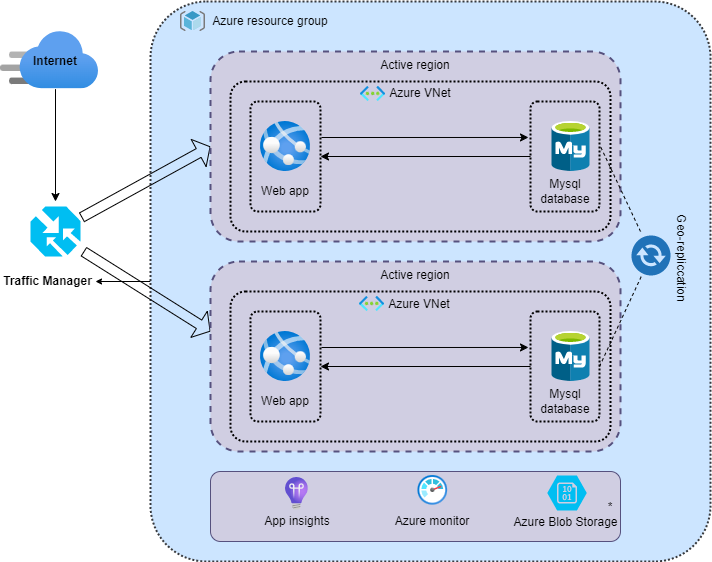


Figure 1.1 - Example *Solution Architecture* Overview Diagram

## As-Is Architecture

Drone Shuttles was running their web site on premise, and it was working properly on daily basis until they decided to kick-off their first marketing campaign announcing their new products and offers. That time, a huge number of users visited their web site where it was beyond their expectation and the current infrastructure and resources which they have did not fulfill this number of users which affected their sales and business. Hence, they decided to think about how to solve these demands ASAP.

## To-Be Architecture

According to current architecture limitations, we have decided to move to Microsoft Azure cloud and benefits from their offerings to fulfill all business demands and design something that works on the short and long term with minimum maintenance and operations headache beside optimizing all costs.

## Transition Planning

As this project does not involve many resources and business requirements, we have decided that transition from current state to the cloud will be smooth and straight forward buy just building the new solution offline while the current one is running and replicate the whole data asynchronously which won’t affect running business then will plan for a maintenance window for moving traffic from current system to the new one with minimum downtime possible.

# Architecture Goals and Constraints

The Architecture goals and constraints section provides a description of goals and constraints of the *Solution Architecture*. The following subsections each provide specific architecture goals and constraints.

## Architectural Assumptions and Decisions

The proposed architecture was designed according to agreed assumptions and decisions that were approved by the client as follow:

- The application should be able to scale depending on the load.

- There should be no obvious security flaws.

- The application must return consistent results across sessions.

- The implementation should be built in a resilient manner.

- Observability must be considered when implementing the solution.

- The deployment of the application and environment should be automated.

## Solution Architecture Attributes

Based on business requirements and current limitations of the running solution, it was crystal clear that we need to embrace on essential attributes to the proposed architecture as follow:

* Scalability
* Consistency
* Security
* Operation excellence
* Cost optimization
* Elasticity
* Availability and disaster recovery

So, we have listed above the base attributes for the proposed solution and we are going to discuss in details these attributes according to proposed solution.

### Technology

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Category** | **Front-End** | **Back-End** | **Integrations** | **Storage/Databases** |
| **Software** | Ghost platform | NodeJS by Ghost platform | N/A | MySQL 8.0 |
| **Hardware** | N/A | Azure cloud managed services by Microsoft | N/A | Azure database for MySQL |

Table 2.1 - Hardware and Software Technologies

### Portability

One of the architecture attributes is portability where the client would be able to host the whole solution on different provider if decided or even revert to host it on premise as the web or front and back ends packed as a docker image that could be containerized anywhere while the back-end database could be replicated anywhere as well

### Capacity

One of the governance factors within proposed architecture was infrastructure capacity. But, this was a real pain with current implementation and need more time for planning and deployment as well but moving to the cloud, it was peace of cake as the cloud provider provide many ways where you would be able to easily manage and resource capacity within your environment and our preference was to do this with Hashicorp Terraform (Infrastructure as code) where increasing to decreasing capacity is a matter of running the DevOps pipeline in minutes plus monitoring and managing all resources from the Azure portal and via notifications sent to our operation emails.

### Performance

Performance was an essential attribute considered in our proposed architecture. Hence, we decided to go with all serverless services offered by Microsoft Azure for both front and back ends including the database storage.

Serverless services empower our infrastructure with enough capacity up to needed load in off-peak and peak hours on daily basis and this is done automatically and could be managed manually as well if needed.

### Availability and Reliability

Availability and reliability were too one of our architecture attributes where we empowered Microsoft Azure services to enable redundancy to solution components like Azure app service and MySQL database.

Azure services offer a good SLA plan as well according to business and operations needs

### Scalability

Scalability too was very important factor to our proposed solution as the client was expecting increasing volume f users during marketing campaigns, so we have empowered the capabilities of Microsoft Azure services to upscale to downscale our resources according to users load through monitoring mechanism to servers’ resources like CPU and memory.

### System Management, Monitoring, and Administration

As the new proposed architecture will be hosted on Microsoft Azure. So, majority of infrastructure components will be managed automatically by Microsoft, and they provide a web portal and CLI tools to let you control and manage everything in few clicks.

Also, we have utilized Azure application insights and monitoring services to keep the client updated with system statuses and alerts ahead of time to avoid any unexpected crises.

### BC/DR and COOP

As one of the business requirements, we have considered a good business continuity plan to the proposed system by designing a full-fledged replica from all resources within the current region to another different standby region to ensure business continuity in case of unexpected disaster or unavailability of the active region or the operational one and we have used Azure traffic manager to direct all users traffic to the standby region if any failure occurred.

### Security

As security was one of the most important business requirements, we have decided to go with the basic and essential security practices as the solution was just a blogging solution and does not incur any sensitive information or data, but we decided to follow the iteration approach to apply additional security practices if needed for the current POC.

### Cost Optimization

*One of the great decisions we had made was moving to the cloud as this will tremendously decrease all costs around this solution as follow:*

* *Less IT specialists needed to manage the whole solution*
* *No procurement for additional resources as Azure cloud enables you to pay as you go for consumption only*
* *Disaster recovery site will be deployed if needed and will be in deallocated status until you decide to allocate it*
* *No licenses or additional third-party tools needed for managing and monitoring your infrastructure is this already provided by Microsoft Azure cloud*

## Deployment

*The whole new proposed architecture was designed emphasizing deployment automation into account and one of the major considerable architecture goals and business requirements as well.*

*So, we have implemented the whole infrastructure in code that can be attached to a DevOps pipeline.*

*Not only this, but we have applied the green/blue deployment approach to ensure zero downtime to current running system through different iterations after every sprint end.*

*A separate GitHub repository will be provided with the POC including all infrastructure terraform code needed for the deployment automation.*