**Solution Requirements**

Based on the business case and acceptance criteria, the solution requirements are as follows:

1. The solution must be built on a public cloud platform (In our case AWS)
2. The solution must use the Ghost Blog platform for the customer's marketing efforts.
3. Scalable: The solution must be able to handle traffic spikes, with the ability to scale depending on the load. The solution can easily scale up or down based on traffic load.
4. Resilient: The solution must be designed in a resilient manner to ensure high availability even in case of a significant geographical failure. The solution is designed to remain online even in the case of significant geographical failure.
5. Disaster recovery capabilities: The solution must have disaster recovery capabilities in case of a region failure. With Amazon RDS, it's easy to create a read replica in another region for disaster recovery.
6. The solution must support multiple separated environments to support the customer's development efforts.
7. The solution must enable the DevOps teams to release new versions of the application multiple times per day without requiring any downtime.
8. Observability & Monitoring: The solution must provide tools to support the customer's operations and help them with visualizing and debugging the state of the environment. The solution must ensure observability by considering metrics, logging, and tracing. The solution includes monitoring with Amazon CloudWatch to ensure that the application is running smoothly, and any issues are quickly identified.
9. Security: The solution must provide visibility to the security team into the platform and its operations. The solution can be made secure by configuring appropriate security groups, VPCs, and access controls.
10. Cost-optimized: The solution must be optimized for costs and easy to maintain/develop in the future. Using AWS services means that the solution is cost-effective and can be easily optimized as needed.
11. The solution must enable the customer to delete all posts at once using a serverless function.

**Solution Design**

Based on the requirements and acceptance criteria provided, I propose the following solution architecture for Drone Shuttles Ltd:

1. Cloud Platform: Amazon Web Services (AWS)
2. Ghost Blogging Platform: Ghost Blog

The Ghost Blogging Platform: This is where the Ghost application will be installed and run. It will be deployed in a scalable and resilient manner in AWS cloud environment. The platform will be accessible via the internet and secured using appropriate security measures. Amazon Elastic

1. Deployment Method:

We will create a Compute Cloud (EC2) instance to host the application. EC2 instances can be quickly and easily created, modified, and terminated based on traffic load. (EC2 Instance or (Docker))

1. Load Balancer: Elastic Load Balancer (ELB)

The load balancer distributes incoming traffic to the available instances of the Ghost application. We will use the Elastic Load Balancing service from AWS to create a load balancer that will distribute traffic to multiple instances of the Ghost application. This will help to distribute the load during high traffic periods.

1. Auto Scaling: Auto Scaling Groups (ASG)
2. Disaster Recovery: Multi-AZ deployment

In case of a region failure, the solution will have a disaster recovery plan to ensure minimal to no downtime. The disaster recovery plan will involve replication of data across multiple regions and a well-defined failover procedure.

1. Amazon S3:

This is where we will store media files, such as images and videos, that are uploaded to the Ghost application. Amazon S3 is a highly scalable and durable object storage service that can be used to store and retrieve any amount of data.

1. CI/CD: CodePipeline, CodeBuild, and CodeDeploy
2. Amazon CloudWatch

Monitoring: Observability The solution will be designed to provide comprehensive monitoring and observability features. Metrics, logs, and traces will be collected and stored in a centralized location for easy analysis and debugging. The system will use automated alerts to notify the DevOps teams in case of any anomalies or errors. This is a monitoring service that will be used to monitor the health and performance of the EC2 instances, load balancer, and RDS instance.

1. . Amazon CloudFormation

This is an AWS service that provides a way to create and manage a collection of related AWS resources, potentially across multiple regions and accounts, in an orderly and predictable fashion

1. Serverless Function: AWS Lambda

will be created to provide the ability to delete all posts at once. Access to the function will be secured and audited to prevent unauthorized access.

1. Security: AWS Identity and Access Management (IAM)

Appropriate security measures will be implemented to ensure the security of the platform and data. Access to the platform will be secured using multi-factor authentication, SSL encryption, and network security groups. Data at rest and in transit will be encrypted using appropriate encryption methods.

1. Cost Optimization

The solution will be optimized for cost by leveraging the use of on-demand and spot instances. The usage of resources will be monitored and optimized to ensure minimal waste and optimal performance.

1. DevOps Teams

Five DevOps teams will be given access to multiple separated environments to support their development efforts. The environments will be created using AWS Infrastructure and will be automated for efficient and consistent deployment. The teams will be able to release new versions of the application multiple times per day, without requiring any downtime.

1. AWS Systems Manager:

This is an AWS service that provides a unified user interface for managing your resources across multiple AWS accounts and regions. We will use this to manage the EC2 instances.

**Architecture Overview**

The proposed architecture consists of multiple components, which work together to create a highly available, scalable, and secure blogging platform.

The Ghost Blog platform will be deployed using Docker, which will enable easy management and scaling of the platform. The platform will be deployed in multiple Availability Zones (AZs) to ensure high availability and disaster recovery.

The Ghost Blogging Platform will be deployed in an Auto Scaling Group (ASG) behind an Application Load Balancer (ALB) to provide scalability and load balancing. The ASG will be configured to launch instances in multiple AZs for high availability.   
Elastic Load Balancer will distribute incoming traffic across multiple instances of Ghost Blog running in the Auto Scaling Groups (ASGs). The ASGs will be configured to automatically scale the number of instances based on the traffic load.

The data for the platform will be stored in a managed database service provided by the cloud provider, which will also be replicated across multiple AZs for durability and resilience.

The CI/CD pipeline will be implemented using AWS CodePipeline, CodeBuild, and CodeDeploy. The pipeline will enable automated testing, building, and deployment of new versions of the application.

Monitoring and observability of the application will be implemented using Amazon CloudWatch. The monitoring system will provide real-time metrics and logs for the application, allowing the DevOps teams to quickly identify and resolve any issues.  
Metrics, logs, and traces will be collected and stored in a centralized location for easy analysis and debugging.

To enable the deletion of all posts at once, a serverless function will be implemented using AWS Lambda. The function will enable the security team to quickly delete all posts from the application.

Finally, security of the platform will be implemented using AWS Identity and Access Management (IAM). IAM will enable fine-grained access control to the platform, ensuring that only authorized users can access sensitive data.  
Security measures will be implemented at multiple levels, including network security groups, SSL encryption, multi-factor authentication, and appropriate access controls.

Cost optimization will be achieved by leveraging the use of on-demand and spot instances, and by monitoring and optimizing the usage of resources.

**Deployment:**

1. Create an Amazon RDS instance to host the Ghost database.
2. Create an Amazon S3 bucket to store media files.
3. Create a new Amazon EC2 instance, install and configure Ghost.
4. Create a new Elastic Load Balancer (ELB) to distribute traffic to the EC2 instances.
5. Configure Amazon Route 53 to route traffic to the ELB.
6. Set up Amazon CloudWatch to monitor the EC2 instances, ELB, and RDS instance.
7. Use AWS Systems Manager to manage the EC2 instances.
8. Use AWS Lambda to provide the ability to delete all posts at once.
9. Use Amazon CloudFormation to automate the deployment process.

**Conclusion**

The proposed solution meets all the acceptance criteria and addresses the business case requirements of Drone Shuttles Ltd. This proposed architecture will enable Drone Shuttles Ltd to launch a highly available, scalable, resilient and secure blogging platform using the Ghost Blog platform. The solution will also provide disaster recovery capabilities, enable easy management and scaling of the platform, and provide a robust CI/CD pipeline. The use of serverless functions and monitoring tools will also enable the DevOps teams to easily manage and debug the platform. It also provides comprehensive monitoring and observability features, disaster recovery capabilities, and security measures to ensure the security of the platform and data. The solution also provides cost optimization.

**Documentation:**

Comprehensive and understandable documentation of the solution will be provided, which includes:

* Architecture diagrams and explanation
* Detailed deployment instructions using the AWS Infrastructure
* Monitoring and observability instructions
* Disaster recovery plan and procedures
* Security measures and access controls
* Cost optimization strategies
* Serverless function instructions