Requirement

2. Infrastructure

You are tasked with setting up a kubernetes cluster to host new web services. The web services will be providing a public API to an existing internal systems. The internal systems are hosted on a virtual network named "internal-assets", and it is imperative that a high level of security is maintained around this virtual network.
Please provide a brief description of the infrastructure you would put in place. You can describe your set up in words or pictures, but please ensure you account for the following constraints:

1. You have a choice of which hosting provider it is deployed with.
2. Your cluster is going to host web services that need to be published on the internet.
3. A support team will need to be notified if web services lose connectivity to the internal assets
4. Developers should be able to deploy code in an automated manner.
5. The cluster will need to be able to access pre-existing internal systems on the "internal-assets" virtual network. Describe how we can access that securely.

You do not need to provide complete implementation details for the above, but please be specific about the selection of technologies you would use.

The solution

Infrastructure Overview

- **Hosting Provider**

  Azure Kubernetes Service (AKS): Azure is chosen for its integration with various Azure services.

- **Cluster Setup**

  AKS Cluster: Create an AKS cluster to host the web services.

- **Public API Access**

  Azure Application Gateway with Web Application Firewall (WAF): Use Azure Application Gateway to publish the web services on the internet. It provides load balancing, SSL termination, and protection against common web vulnerabilities with WAF.

  Azure Front Door: Optionally, use Azure Front Door for global load balancing and enhanced security.

- **Security for Internal Systems Access**

Virtual Network Peering: Peer the AKS virtual network with the "internal-assets" virtual network. This allows secure, low-latency communication between the AKS cluster and internal systems.

Network Security Groups (NSGs): Implement NSGs to restrict traffic between the AKS cluster and the internal-assets network to only the required ports and IP addresses.

Private Link: Use Azure Private Link to access Azure services privately over the Microsoft backbone network, ensuring data never traverses the public internet.

- **Monitoring and Alerts**

  Azure Monitor and Azure Log Analytics: Set up Azure Monitor to collect logs and metrics from the AKS cluster. Use Azure Log Analytics to analyze logs and set up alerts.

  Azure Alerts: Configure alerts to notify the support team if connectivity between the web services and internal systems is lost.

- **CI/CD for Automated Deployment**

  Azure DevOps or GitHub Actions: Use Azure DevOps pipelines or GitHub Actions to automate the deployment of code to the AKS cluster.

  Helm: Use Helm charts to manage Kubernetes application deployments, making it easier to version and roll back changes.