Executive Summary
Introduction
Blue Ridge Global is a software company offering supply chain planning solutions, with advanced algorithms and machine learning. They faced the primary challenge of scaling their infrastructure to accommodate a growing customer base, while also addressing maintenance issues and downtime. To overcome this, an AWS Migration solution was implemented, leveraging cloud scalability, fault tolerance, high availability, and cost savings.

Customer Challenges
Blue Ridge Global encountered scalability, infrastructure obsolescence, maintenance and upgrades, inadequate monitoring systems within their existing infrastructure. Their foremost concern was scalability, driven by a remarkable 50% growth in their customer base in the past year. With the infrastructure nearing obsolescence, a complete upgrade demanded a substantial capital expenditure of $500,000. They also faced frequent maintenance and upgrade issues, leading to an average monthly downtime of 10 hours. The lack of effective monitoring systems hindered the identification of root causes promptly. This lack of observability impeded swift issue resolution. To accommodate their expanding customer base, they aim to transform their software into a SaaS offering, anticipating a 20% increase in revenue.

Our Solution

The application was moved in 2 different phases to achieve the business goals.

1) **Migration Phase**
   - Lift and shift migration method was implemented for their existing legacy .NET application
   - Amazon EC2 instances are considered in place of the on-premises VMs
   - IIS and other .Net dependencies are installed on the Amazon EC2 instance
   - Leveraged the AWS Database Migration Service to sync and migrate the data on MS SQL running on Amazon EC2

2) **Modernization Phase**
   - Migrating ASP. NET-based web applications configured on IIS and Windows scheduled tasks from Amazon EC2 Windows server instances to containerized architecture on Amazon EKS
   - Converting the existing multi-tenant web application into multiple single-tenant web application instances, analyzing the resource consumption of each tenant, and providing custom features to each tenant
   - Nginx ingress backed by AWS NLB is implemented for traffic management and establishing auto-scaling strategies for optimal resource utilization and high availability by leveraging AWS Autoscaling with Amazon CloudWatch policies based on CPU and memory metrices

About Blue Ridge Global

Blue Ridge Global, headquartered in Atlanta, Georgia, is a software company specializing in business supply chain planning solutions. Their advanced algorithms and machine learning techniques optimize inventory management, demand forecasting, and pricing strategies. They serve customers worldwide across various industries, providing solutions for demand planning, inventory optimization, price optimization, and replenishment planning.

- Amazon CloudWatch is set up to analyze custom metrics, and access to real-time custom application logs is provided for faster troubleshooting.
- We have implemented continuous integration and continuous deployment integrating Bitbucket, ECR, TeamCity, Octopus on Amazon EKS
- Created and configured Amazon EKS cluster on Linux and Windows nodes with cluster autoscaling along with AWS Load Balancer Controller and Nginx Ingress Controller to expose applications running in a Amazon EKS for external access
- Establish a single deployment for the login application and expose it using an AWS Application Load Balancer.
- Set up a dedicated Kubernetes Deployment resource per tenant for all other applications, exposing them using a cluster-IP service type.
- Enable HPA based on CPU and memory consumption for application deployments
- Implement infrastructure and application monitoring using Amazon CloudWatch and integrated Datadog with Amazon EKS to access container logs from the Datadog Dashboard.
- The AWS Services leveraged in the solution are AWS CloudFormation, Amazon Elastic Kubernetes Service (EKS), Amazon EC2, Amazon CloudWatch, Amazon Elastic Container Registry (ECR), AWS Elastic Load Balancing.
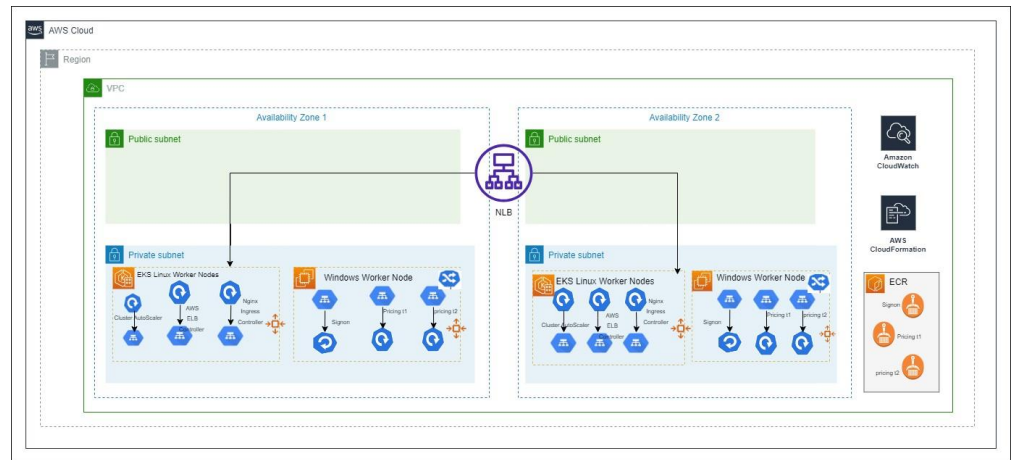
Architecture Diagram

Fig 1: Migration and Modernization process of an applications containerized on EKS

Conclusion

• Pods within the cluster improved the application's startup time, transforming it from 5 minutes to 5-6 seconds, resulting in a 99% improvement.

• Implementing a Rolling Update strategy in Kubernetes deployment reduced the tenant onboarding downtime from 3-4 minutes to less than a minute, resulting in a 75% reduction in downtime.

• Automated infrastructure provisioning with eksctl and CloudFormation and allowing for a streamlined process.

• Efficient use of HPA and cluster autoscaler allowed to smoothly scale resources based on the criteria, reducing resource wastage, and saving the company a remarkable 10% in costs while allowing the customer to scale and maintain availability.

• With the integration of CI/CD and DevOps best practices, new feature releases and rollbacks could be achieved with zero downtime, providing users with a seamless and uninterrupted experience.