Tech Mahindra

Enterprise application modernization with Azure Kubernetes Service on Microsoft

WHITEPAPER

Connected World.
Connected Experiences.

Abstract

Most enterprises have significant investments in their existing application portfolio, from both a financial and operational standpoint. While the term “legacy” sometimes has a negative connotation in the enterprise software library, these legacy systems are often most mission-critical applications from business perspective.

A related modernization trend is the IT industry embracing containers and orchestration as a means for packaging, deploying, and managing applications and workloads on cloud. While we can containerize a legacy app, containers are viewed as an optimal fit for a more decoupled approach to development and operations—namely, microservices based architecture.

Introduction

Today’s businesses are faced with a singular reality: innovation is the requirement for mere survival. Yet many enterprises are crippled by legacy and technical debt. This paper is written for the leaders tasked with bridging the gap. Cloud’s approach to modernizing infrastructure, process, and architecture (IPA) equips senior IT decision-makers with a realistic, achievable path to application modernization. In this paper, you’ll learn how to tackle seemingly insurmountable challenges one step at a time and see real-world examples from enterprises that have already succeeded.

The goal of traditional application modernization is to optimize both the velocity and the efficiency of an application’s release cycle. By introducing new technologies and embracing new processes, businesses can deliver value more quickly. Innovation velocity is the speed at which a team can introduce something new and of value to its customer at a reasonable cost and is often tied to technology strategies and choices. Process efficiency is the ability for a team to improve how it brings innovation to market with the least amount of friction and is often related to process methodologies such as information technology infrastructure library (ITIL), Waterfall, DevOps, or Agile. The client is one of the global leaders in the logistics industry. With more than 300,000 people in over 200 countries, it has delivered more than a billion parcels worldwide. This organization needed to transform some of their legacy enterprise application into microservice based application and deploy those applications to cloud.

Industry Landscape

Application modernization is a strategic decision factoring in organizational needs, priorities, and budgets. The considerations include modernizing the application experience, method of accessing, and creating new workflows around the application through integration and automation.

Traditional applications limit an enterprise’s ability to move quickly in two ways. First, the monolithic architecture of a traditional application is inherently inflexible, creating exponential inefficiencies when building and running applications. Second, traditional applications constrain development for new, cloud-native applications that depend on them. Application components in monolithic architecture are tightly coupled; changes to any individual component requires changes to other components.

Key Takeaways

- App modernization on Azure
- Containerization with microservices based architecture on Azure Kubernetes Service
- Release lifecycle management with Azure native CI/CD platform i.e., Azure DevOps
- Infrastructure automation with Infra-as-a-Code tool Terraform
- Security
- Holistic approach
- Integration with 3rd party analytics tools

The Key Focus Areas Include

Application modernization strategies or plan

Planning on Application migration to cloud

Breaking monolithic applications to microservice based applications

Assist in containerizing these microservice based applications

Delivering apps and features faster with containers and container orchestration

Implement DevOps lifecycle

Provide a stable environment and seamless operations support

Enhance security

Enable intelligent security analytics and threat intelligence

Reducing the overall Total cost of ownership

Benefits

These services help the businesses to:

Reduced total cost of ownership (TCO)

Better agility, scalability and portability

Improved security with Azure's distributed denial-of-services and threat protection

Better performance from highly available application. Less downtime

Faster time to market from weeks to days

Reduction in IT admin cost due to the self-service module implementation to provision the resources

Improved productivity due to less operational activities and more focus on the app deployments rather than managing the cluster

Our Solution

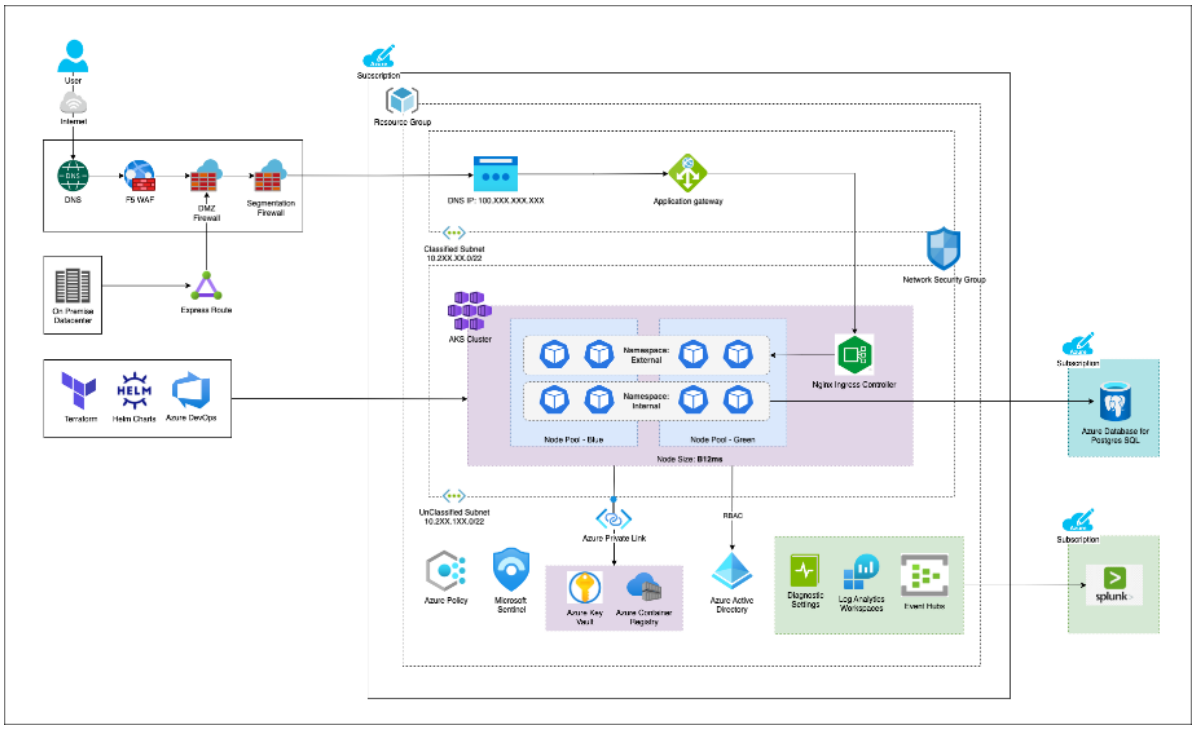
The services which are offered in the solution are-

- Establish right cloud strategy as per the business needs
- Evaluate impact of issues related to security, governance, risk & compliance
- Define a cloud-based application modernization strategy
- Implement the project
- Provide managed operation services:
    - 24/7 operation & monitoring of the platform by our support engineers
    - Providing L1, L2 and L3 supports
    - Escalation to Microsoft support team
    - Managed blue/green deployment process
    - Updating and maintaining the workload templates

Simplified Modern Architecture

This app modernization solution has been designed and built considering Microsoft best practices guidelines and with Microsoft Azure Well-Architected framework standards. This solution includes autoscaling functionality applied both on cluster and pod level. These applications are hosted in AKS cluster in Azure East US region that spread across 3 availability zones to provide high availability.

System Design



Components Of The Architecture

Azure Active Directory (Azure AD): It is an enterprise identity platform provides single sign-on and multi-factor authentication to govern user access to resources.

Azure Resource Group: Resource groups are used to group Azure resources so they can be managed by lifetime, owner, or other criteria.

Azure Kubernetes Service (AKS): It is a managed Kubernetes service with enterprise-grade security and governance and offers platform to deploy and manage containerized applications. It is based on the open-source Kubernetes (k8s), a container orchestrator tool, which is designed to build, deliver, and scale containerized applications. The Kubernetes API server is managed by Azure.

Azure Pipelines: It is a service that provides Continuous Integration and Continuous Delivery jobs, to build and release your application automatically.

Azure PostgreSQL: It is a fully managed PaaS database engine that handles most database management functions like upgrading, patching, backups, and monitoring, without user involvement.

Azure Monitor: It lets you get insights on the availability and performance of your application and infrastructure. It also gives you access to signals to monitor your solution's health and spot abnormal activity early. It can collect log data from VM operating systems as well as crash dump files and aggregate them for viewing in Microsoft Defender for cloud.

Microsoft Sentinel: It is a cloud native security information and event management (SIEM) and security orchestration, automation and response (SOAR) solution. It uses advanced AI and security analytics to detect, hunt, prevent, and respond to threats across enterprises.

Virtual network (VNET): It provides an isolated and highly secure application environment by restricting network access to specific IP addresses or subnets. By default, AKS creates a virtual network into which agent nodes are connected.

Azure Private Link: It provides a private endpoint in a Virtual Network for connectivity to Azure PaaS services like Azure Storage and SQL Database, or to customer or partner services.

Azure Container Registry: It hosts your docker container images. This service includes container image scanning with the integration with Microsoft Defender for cloud.

Azure Policy: It lets you create, assign, and manage policies. These policies enforce different rules and effects over your resources, so those resources stay compliant with your corporate standards and service level agreements. It integrates with Azure Kubernetes Service too.

Microsoft Defender: It is for Endpoint protects organizations from threats across devices, identities, apps, email, data, and cloud workloads.

Log Analytics: It is a monitor service that you can use to query and inspect monitor log data. Log analytics also provides features for charting and statistically analysing query results.

Azure Key Vault: It securely stores and tightly controls access to secrets like API keys, passwords, and certificates.

Terraform: It is a third-party tool that provisions and modifies resources per environment. It also supports cross-platform infrastructure-as-code configuration and deployment across Azure and other cloud providers.

Helm: It is a package manager for Kubernetes. Helm is the K8s equivalent of yum or apt.

Detailed Approach for Implementing the Architecture

An Azure subscription has been linked to the client's HUB account in Azure Active Directory (AD) Connect. Within that subscription, Azure Kubernetes Service (AKS) resources and related Azure resources got provisioned. This helps us to maintain a secure environment.

- A resource group in that Azure subscription has held those related resources for AKS Azure solution. This resource group includes all the resources for the solution for ease of management and billing purposes.
- AKS clusters have been created with Azure Container Networking Interface (CNI), so that every pod gets
- These IP addresses are planned to be unique across the network landscape. Each node has a configuration parameter for the maximum number of pods that it supports. The equivalent number of IP addresses per node have been then reserved up front for a particular node.
- The cluster node pool sizing is arrived with the best practices to serve the below purposes

High availability

Autoscaling

Capacity management

Performance of the application hosted in the cluster.

- The AKS cluster deployment is an automated process, and we can create/update these clusters with Terraform scripts.
- The cluster autoscaler has been enabled to let us run an efficient, cost-effective cluster.
- When using Azure CNI, every pod is assigned a Virtual Network route-able private IP from the subnet. So, the gateway can reach the pods directly.
- Azure Network Security Group (NSG) has been used for the traffic firewall to the cluster and between the node pools, additionally all the microservices and pods are under strict network isolation utilizing Azure CNI-Calico security policies.
- Application Gateway has been used at the overall cluster level and it will be in the classified segmentation. First Service reached through application gateway must be in the external segmentation zone.
- Ingress load balancers and their respective rules were provisioned during Deployments as per the requirements, Ingress controller Load balancer got created as Internal.
- AKS cluster requires certificates which are stored in Azure key vault.
- Terraform state persistency is managed in Azure storage Account as terraform backend.
- Azure Container Registry (ACR) with premium SKU created for this project to store the images relevant for AKS cluster. ACR got provisioned through Azure DevOps pipeline using Terraform modules.

- Azure AD group membership was used to control access to namespaces and cluster resources using Kubernetes role-based access control (RBAC) in an AKS cluster.
- Role assignments scoped to the entire AKS cluster was done on access control (IAM) blade of the cluster resource using Azure CLI command
- Cluster Role and Cluster role binding got created for the below said AD groups in cluster level

Role 'Azure Kubernetes Service RBAC Reader' got assigned to 'Monitoring Team – Read Only'Autoscaling

Role 'Azure Kubernetes Service RBAC Admin' got assigned to 'Application Developer (Elevated Access)'

Role 'Azure Kubernetes Service RBAC Writer' got assigned to 'Application Developer'

Role 'Azure Kubernetes Service RBAC Cluster Admin' got assigned to Cluster Admins'

- Container Storage Interface (CSI) driver got installed in the AKS cluster through Helm repo (as part of the cluster creation pipeline). The purpose of the CSI driver is to access and retrieve secrets from key vault using the secrets store Container Storage Interface (CSI) driver to mount the secrets into Kubernetes pods.
- Resource lock got applied to the resources to prevent resource deletion by accident
- Azure Defender is enabled for container registries at the subscription level
- Enhanced security Kubernetes secrets by storing them externally in Azure key vault
- Used security hardened VM host image to reduce attacks
- Enabled Microsoft Sentinel to deliver intelligent security analytics and threat intelligence across the enterprise, providing a single solution for attack detection, threat visibility, proactive hunting, and threat response
- For monitoring and management of the AKS cluster below are the use cases considered

Log Aggregation

Cluster health monitoring

Security and audit

Application Monitoring

- For Splunk integration, we have installed collector for Kubernetes agent in the AKS cluster, so that Splunk can retrieve the logs from cluster.

A well designed, planned, and tested business continuity and disaster recovery strategy is essential to protect a business from planned and unplanned outages.

Availability zones are a high availability offering that protects the applications and data from data center failures. Zones are unique physical locations within an Azure region. Each zone is made up of one or more data center equipped with independent power, cooling, and networking. To ensure resiliency, there's always more than one zone in all zone enabled regions. The physical separation of availability zones within a region protects applications and data from data center failures.

AKS clusters that are deployed using availability zones which can distribute nodes across three zones within a single region of East US.

If an availability Zone 1 (East US DC1) becomes unavailable, the applications will continue to run across availability Zone 2 (East US DC2) or availability Zone 3 (East US DC3).

The solution is not protected against an outage of the whole Azure data center location East US.

NXT.NOW™ Advantage

IP Range Authorization: For public cluster, IP ranges should be configured, so that API server will be accessible only from that ranges.

Storing Secrets in Azure Key Vault: Using Key Vault Secrets for injecting passwords through CSI driver

Isolating Groups of Resources: Using Kubernetes namespaces to properly isolate the Kubernetes resources. We do not use the default namespace / Namespaces

Implement Pod Identity: Using pod identities to automatically request access using a central Azure AD identity solution

Scan the Container Image against Vulnerabilities: To protect the Azure Resource Manager based registries in the subscription, we have used Microsoft Defender for container registries

Allow Deploying Containers Only from Known Registries: To ensure only allowed container images in AKS by built-in Azure Policy

Role-Based Access Control to Docker Registries: To have Azure Container Registry roles and permissions

Network Segmentation of Docker Registries: Assign virtual network private IP addresses to registry endpoints and use Azure Private Link

AAD Integration: Sign in to an AKS cluster by using your Azure AD authentication token

K8S RBAC + AAD Integration: Control access to cluster resources

Compliance enforcement of Docker Image Builds: To have Azure Policy built-in definitions for Azure Kubernetes Service

AKS and ACR integration: Integrate the ACR with AKS cluster to pull images without credentials

Maintain Kubernetes version up to date: Regularly update to the latest version of Kubernetes

Enable master node logs: Enabled Diagnostic settings and streamed the logs to Log Analytics and Event hub

Designed Enterprise AKS Landing Zone: Followed the Microsoft guidelines on creating Landing Zone for AKS

Conclusion

Nowadays, everyone wants to transform their monolithic legacy application to a microservice based application so that their applications can become an agile and scalable product. To respond to the rapid changes in today's world, application must be quick to deploy, always available and easy to maintain.

These microservice based application reduce the application downtime since all the microservices are like individual apps which are loosely coupled to each other. So, maintaining, deploying, scaling, upgrading or even deleting these microservices are very easy to achieve. In this paper, we learnt about how Microsoft Azure can help customers to modernize their monolithic on-premises application and provide a highly scalable platform to deploy those applications. Azure provides better agility and scalability for deploying your application with its improved security and compliance certifications.

Authors

Mr. Guru Prasad C P,
guruprasad.cp@techmahindra.com
Group Practice Head, Azure Cloud Services

Guru Prasad C P has an experience of over 22 years with over 8 years specifically in the public cloud working in Asia, ANZ, Europe and the US. His experience includes, setting up practice teams aligned to industry verticals and horizontals, analyst interactions for positioning the offerings, hiring the right talent, involving in strategic exercise mergers and acquisitions , organization building, creating frameworks & IP's.

At Tech Mahindra he is responsible for practice and competency development which includes alignment with OEMs for solutions, offerings and adoption of new technologies, customer interfacing where he acts as a trusted advisor in providing unbiased views/opinions and aligning with organization goals at the same time, value creation, developing practice areas deal making, solution support for large deals, and carve out deals from azure and hybrid cloud perspective

Guru has a keen interest and expertise in verticals including travel, transport, manufacturing, insurance, educational and government charitable trusts.

Team

Mr. Arunava Basu,
ab00788419@techmahindra.com
Solution Architect (Application Modernization/ SRE/ DevOps/ Automation)

Arunava Basu is a Solutions Architect at Tech Mahindra. He is a seasoned professional with 13 years of experience in architecting cloud native applications, migrations, and administration. He also has experience as a cloud DevOps and automation architect to automate tools infrastructure, CI-CD Platform, application provisioning, deployment management with deep understanding and scaling of DevOps process and tools to build stable products. Arunava has a keen interest in automating things.

Mr. M Rajashekar Reddy,
MX00797486@TechMahindra.com
Solution Architect (Application Modernization/ Integration/ Analytics)

M Rajashekar Redd is a Multi-Cloud Architect with 15 years of experience, leveraging proven product, program management, pre-sales and technical architecture skills. He has an experience in working for different clients across the industry verticals such as aerospace, healthcare, insurance, power and energy, oil and gas and telecom and semiconductors, geo- spatial, transportation. His expertise includes Azure, AWS, GCP, OCI, technical program management, delivery leadership, product management, pre-sales, proposal management, agile project management. He is skilled in Python, Java, Apache-Airflow.

Tech Mahindra

www.youtube.com/user/techmahindra09
www.facebook.com/techmahindra
www.twitter.com/tech_mahindra
www.linkedin.com/company/tech-mahindra
www.techmahindra.com