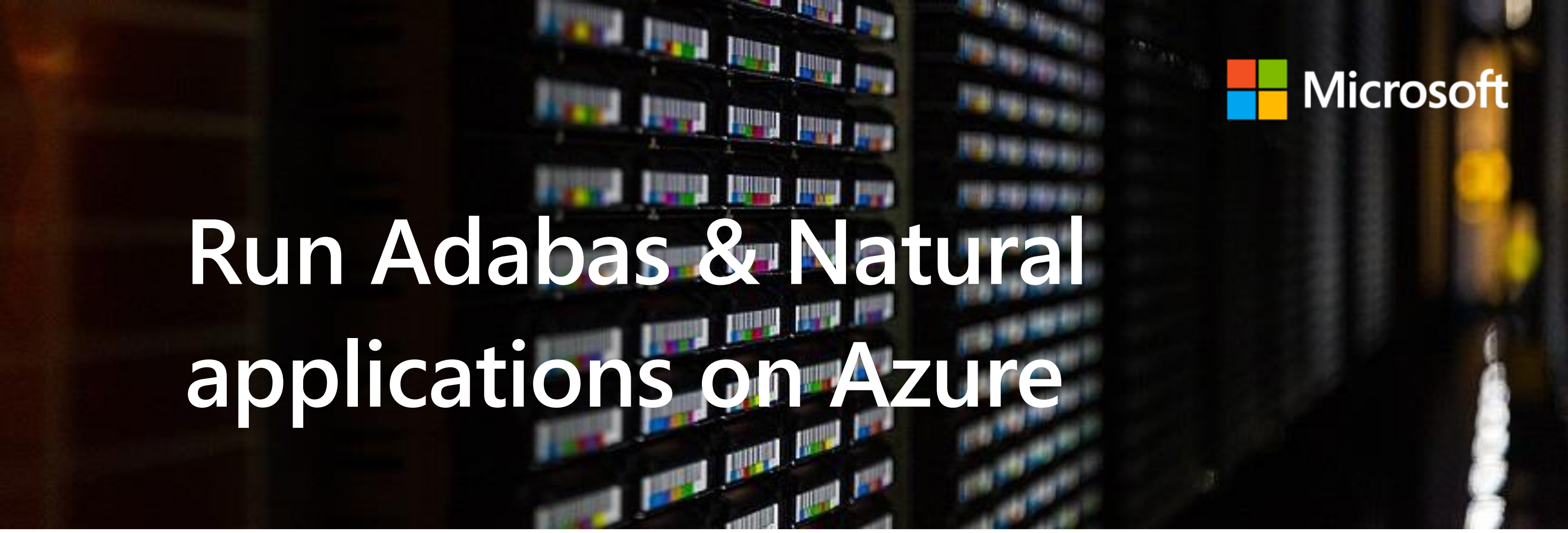

Run Adabas & Natural applications on Azure

Choosing to lift-and-shift or rearchitect

Marlon Johnson
Microsoft Senior Program Manager — Azure Global Engineering

June 2021

For decades, Software AG Adabas has been the adaptable database system behind many large and mission-critical business applications. Now you can bring the convenience of cloud computing to these applications without giving up your Adabas database, the Natural programming language, or even your green screen—unless you want to.

Most organizations we work with are pragmatic in their approach to digital transformation—they want to reuse what they can and make cost-effective choices about the rest. That’s why the lift-and-shift approach to cloud adoption is so popular. You simply move your workload as is, if possible, to Azure Virtual Machines (VMs), a type of infrastructure as a service (IaaS). VMs run in Azure datacenters that are managed by Microsoft, so you benefit from the efficiency, scalability, and performance of a distributed platform without the overhead of hardware management.

However, to gain the full benefits of cloud computing, enterprise application architectures are best rearchitected using modern techniques such as container-based microservices. This type of migration is more complex than a lift-and-shift approach, but the payoff is maximum flexibility and scalability.

This article presents both options and gives you a high-level look at what’s possible whether you keep the green screen or go modern.

Why Azure?

As legacy IT infrastructure can be costlier to operate and maintain, forward-thinking organizations are using Azure. Moving to Azure is a great way to future-proof your software investment without tossing out the valuable Adabas & Natural applications that differentiate your organization.

Azure technologies deliver enterprise-grade reliability, availability, and serviceability. In addition, using Adabas & Natural modernization tools while running on Azure opens the doors to the latest managed services. For example, you can:

- Unlock data to use cutting-edge Azure services for big data analytics, machine learning, artificial intelligence, and high-performance computing.
- Expose APIs for use in communicating with IoT devices and new applications or in modernizing the user experience.
- Improve business continuity with a simple, cost-effective, end-to-end backup and disaster recovery solution that works with your on-premises data solutions.
- Apply agile development practices to Adabas & Natural applications and attract the top talent that expects DevOps.

Based on data from organizations that have moved their core application workloads from legacy IT infrastructure to the cloud, it’s common to see savings in maintenance and operations plus reduced capital costs. To add capacity on Azure, you simply pay for what you need when you need it.

Azure also supports flexible hybrid solutions that enable you to transform at your own pace. You can keep select data and applications on-premises while moving other data and applications to the cloud. If your Adabas & Natural workloads have dependencies on on-premises systems, this type of hybrid operating model is an efficient solution.

As-is IaaS or rearchitect?

For the greatest return on investment, we recommend one of the following cloud transformation options:

- **Lift-and-shift** and run Adabas & Natural applications as-is on Azure Virtual Machines (VMs). These on-demand, scalable computing resources provide the flexibility of virtualization. VMs eliminate the maintenance demands of physical hardware and offer a choice of operating systems. We use Linux in this article for the flexibility, reliability, and cost benefits and that open-source provides.
- **Rearchitect** your Adabas & Natural application using a container-based approach. This allows you to deploy and run the containers in autoscaling clusters orchestrated by Azure Kubernetes System, a free service managed by Microsoft.

Each approach has benefits and tradeoffs, and your particular workload and business case will make the final determination. For example, a lift-and-shift is usually the more cost-efficient option up front, but when you rearchitect, you can reduce significantly reduce long-term costs with on-demand provisioning of resources and other optimization techniques. A lift-and-shift is also quicker than rearchitecting a well-established deployment architecture or even codebase, but the time it takes translates into advantages down the road as the following table shows.

	Lift-and-shift	Rearchitect	
Cost	✓	✓	In the short term, a lift-and-shift is typically less costly but rearchitecting offers long-term benefits.
Risk	✓		Rehosting existing code and apps tends to be less risky overall than introducing new code.
Skills	✓	✓	A lift-and-shift relies on existing skillsets. Rearchitecting requires new coding and DevOps skills but positions organizations to attract top talent going forward.
Range of services		✓	Modernization is sometimes necessary to achieve the full advantages of a cloud platform.
Performance		✓	In rearchitecting, you can optimize apps for the cloud environment in a way that isn't always possible during a lift-and-shift.
Flexibility		✓	Flexible, container-based architectures can be quickly adapted to changing business requirements.
Resilience		✓	By decoupling monolithic components and connecting to highly available managed solutions, your application inherits the durability of the cloud.

Keys to a successful hybrid setup

Whether you choose IaaS or rearchitecting, the network setup between your on-premises datacenter and Azure is one of the first considerations. Azure Virtual Network is a software-defined network that provides logical isolation on Azure. You have complete control over your virtual networking environment. To connect your network to Azure, the simplest option is to set up a virtual private network (VPN). A VPN gateway sends encrypted traffic between the Azure virtual network and your on-premises location over the public internet.

This architecture in Figure 1 subdivides the virtual network address space into subnets. You can associate each subnet with a network security group (NSG) that defines the access policies for the subnet. When associated with a subnet, an NSG offers fine-grained control over resources.

Availability is another key infrastructure decision. For high availability at scale, we recommend using Azure Availability Zones, a configuration that provides datacenter fault tolerance. Availability Zones are physically and logically separated datacenters with their own independent power source, network, and cooling. Connected with an extremely low-latency network, Availability Zones are backed by a 99.99-percent uptime service level agreement for VMs.

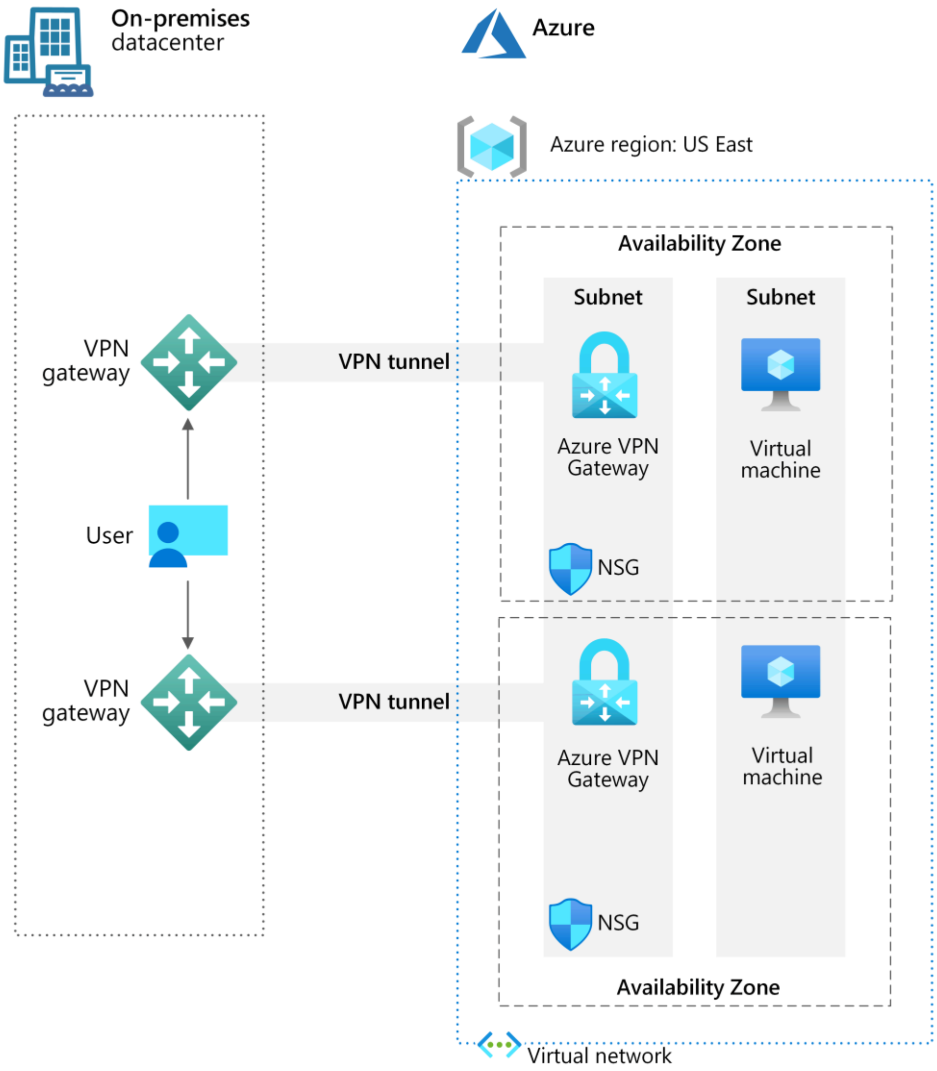

Figure 1. Applications run in multiple VMs in an Azure region. Azure VPN Gateways connect to a customer network over the internet.

IaaS approach: Migrate to virtual machines on Azure

The lift-and- shift approach is a solid first step to gain the benefits of cloud computing, such as cost-effective web hosting, high performance computing, and immense storage resources. Azure IaaS can also help you improve business continuity and disaster recovery with its a global network of datacenters.

For a successful deployment on Azure, you need to consider the size and type of VMs to use and the best storage solution for your workload. As a baseline, consider using F16s VMs, which have 16 cores and support 32 GB of memory. The F-series is based on the 2.4 GHz Intel Xeon E5-2673 v3 (Haswell) processor, which can achieve clock speeds as high as 3.1 GHz with the Intel Turbo Boost Technology 2.0.

In selecting a VM, it helps to consider the MIPS (Millions of Instructions Per Second) rating. We typically use a conservative estimate of 150 MIPS per vCPU. Based on this estimate, for example, an F-series VM with 16 vCPUs could easily support Adabas from a system with 2,400 MIPS.

The simplest lift-and-shift places all components on one VM—a cost-effective solution for workloads that require 200 MIPs or less as Figure 2 shows. All VMs include block-level storage volumes called Azure Managed Disks, which work well for many Adabas & Natural applications.

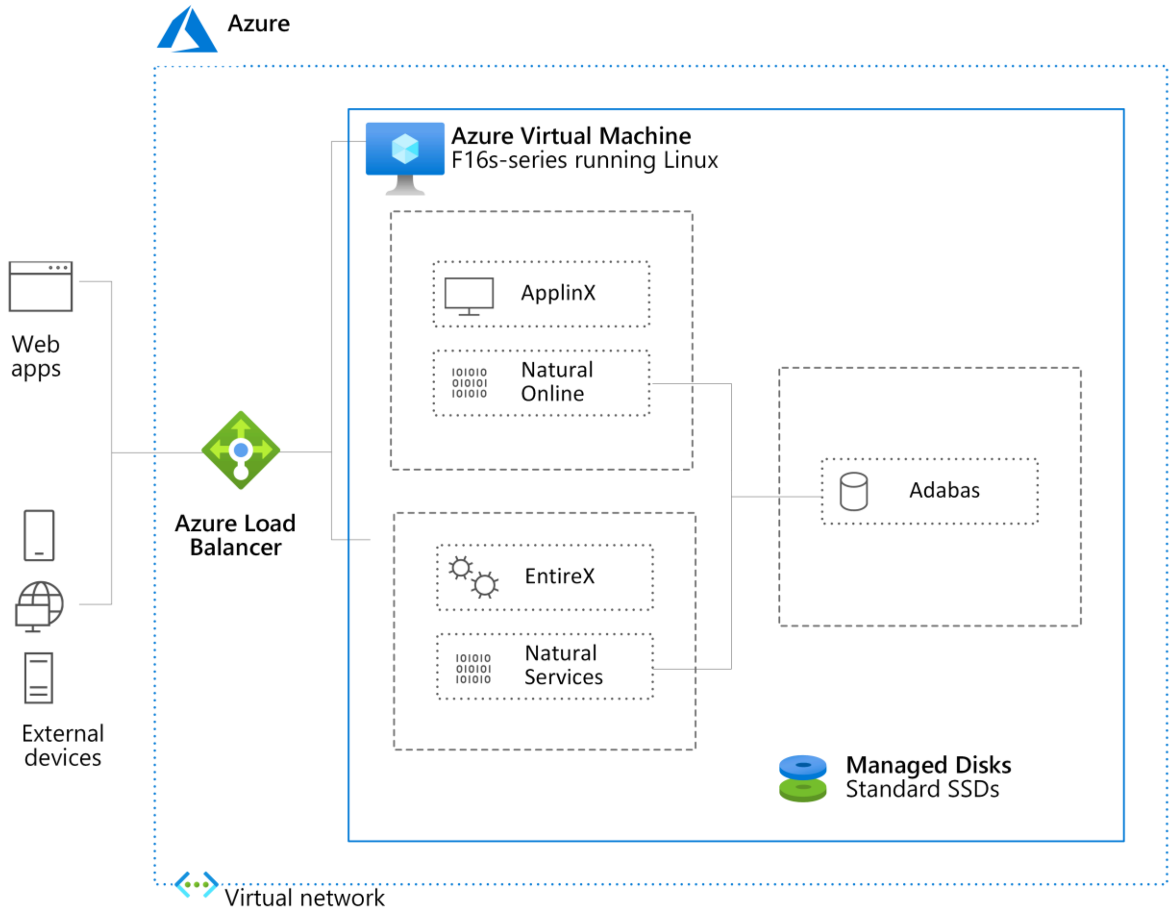

Figure 2. A lift-and-shift migration to a single-instance architecture running on one virtual machine.

In this configuration, the Software AG webMethods ApplinX is installed on the same VM with Natural Online so you can give users the same familiar experience. However, our recommendation is to modernize the user interface. The easiest way to do this is to expose Natural Services as APIs using Software AG webMethods EntireX. Exposing APIs gives you an interface that external sources can call, such as mobile applications, web portals, and IoT devices. If you want to modernize your application's user interface, this is the way to go.

Regardless of the front-end interface components you use, Azure Load Balancer provides a single point of contact for clients. Load balancing evenly distributes incoming network traffic. Azure Load Balancer routes traffic to the healthy instances of Natural, a process that helps you create highly available services.

If your workloads requires more than 200 MIPS, a best practice is to install Adabas & Natural on separate Linux VM instances as Figure 3 shows. Each VM runs in its own subnet, enabling you to use *network security groups* (NSGs) to filter traffic and define security rules for each tier.

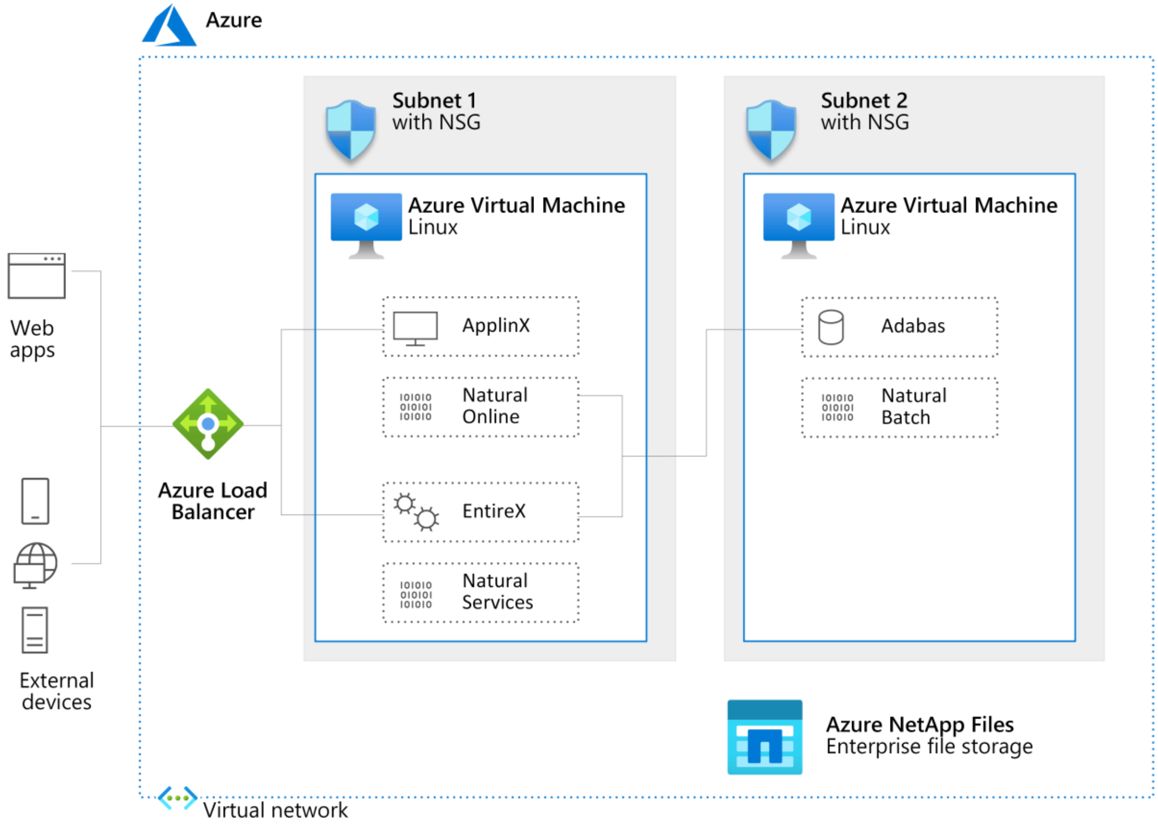

Figure 3. A lift-and-shift migration where the interface components run on one VM while the database runs on a separate VM for scalability.

For high transaction workloads, you can also run Natural and the interface components on *virtual machine scale sets*, a group of load balanced VMs that automatically scale up or down on demand or on schedule. Virtual machine scale sets make it easy to build large-scale services that require high availability.

Natural communicates with the Adabas instance via TCP/IP. For workloads that perform batch transactions, you can install the Natural Batch component on the same VM with Adabas. Natural Batch communicates constantly with the database, so this setup helps reduce latency by keeping the components in close proximity.

Figure 3 also shows the use of Azure NetApp Files. In high-performance scenarios, NetApp Files provides greater throughput than Managed Disks for high volumes, enabling you to run performance-intensive and latency-sensitive file workloads.

Rearchitecting: Migrate to containers and AKS

To make the most of Azure flexibility, reliability, and future capabilities, you have to rearchitect your application. We recommend rewriting monolithic applications as microservices and using a container-based approach to deployment. A container bundles all the software needed for execution into one executable package and includes an application's code together with the related configuration files, libraries, and the dependencies required for the app to run. Containerized applications are quick to deploy and support popular DevOps practices, such as continuous integration (CI) and continuous deployment.

Adabas & Natural containers run in *pods*, each of which is focused on a task. Pods are units of one or more containers that stay together on the same node and share resources such as host name and IP address. Decoupled from the underlying platform, components in pods scale independently and support higher availability. A containerized application is also portable—it runs uniformly and consistently on any infrastructure.

Containerized services and their associated networking and storage components need to be orchestrated and managed. We recommend Azure Kubernetes Service (AKS), a managed Kubernetes offering that automates cluster and resource management. You designate the number of nodes you need, and AKS fits your containers onto the right nodes to make the best use of resources. AKS also supports automated rollouts and rollbacks, service discovery, load balancing, storage orchestration, and self-healing—if a container fails, AKS starts a new one. In addition, you can safely store secrets and configuration settings outside of the containers.

Figure 4 shows a container-based implementation of Adabas & Natural. When you set up AKS, you specify the Azure VM size for your nodes, which defines the storage CPUs, memory, and type available, such as high-performance solid-state drives (SSDs) or regular hard disk drives (HDDs). In this example, Natural runs on three VM instances to boost scalability and availability of the user interface (Natural online plus ApplinX) and the API layer (Natural services plus EntireX).

In the data layer, Adabas runs in the AKS cluster, which scales out and in automatically based on resource use. Multiple Adabas containers can run in the same pod or, for greater scale, AKS can distribute them across multiple nodes in the cluster. Adabas uses Azure Files Storage for all persistent data such as database files, protection logs, app data, and backup. Depending on your requirements, you can also use Azure NetApp Files, a high-performance, metered file storage service.

In this diagram, a batch processing component runs in close proximity to Adabas on the same VM. For larger workloads, batch components can run on a separate VM optimized for performance and cost. On top of this, you can add VMs to run the UI apps.

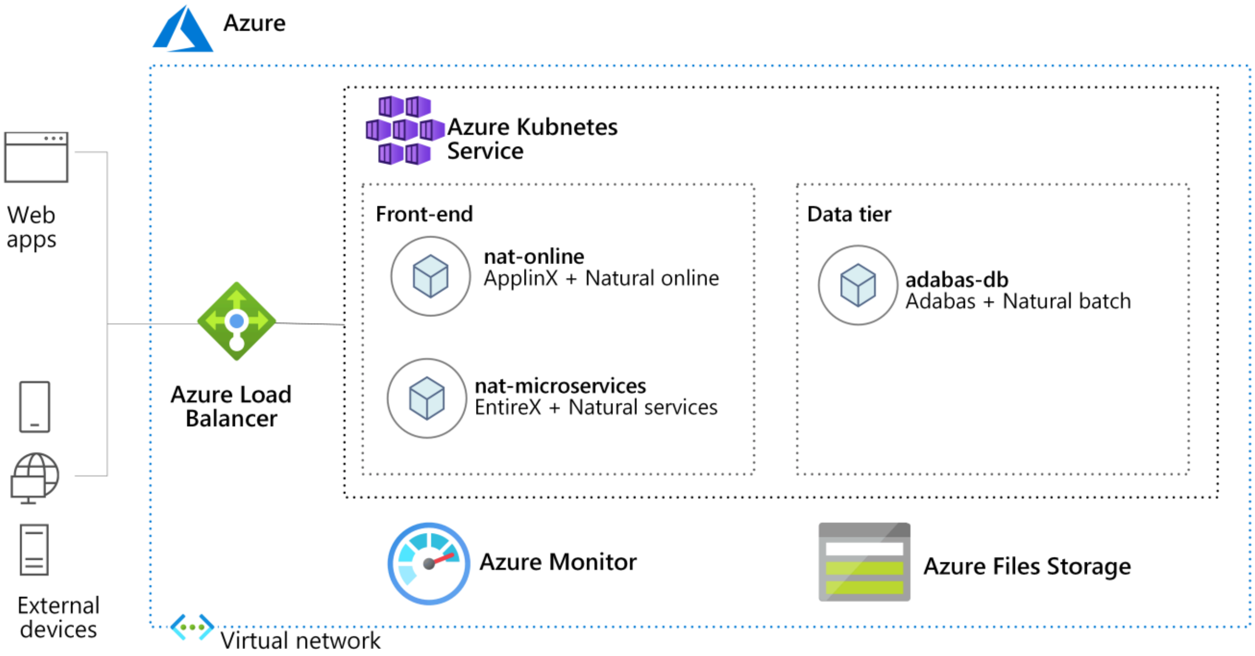

Figure 4. Architecture based on containers and AKS

Azure benefits for Adabas & Natural applications

Whether you opt for the lift-and-shift approach or choose to rearchitect your Adabas & Natural applications, Azure provides the following benefits:

- **Faster deployments through DevOps and CI/CD**. When you rearchitect Adabas & Natural applications to run on Azure, you can use modern DevOps deployment practices and CI/CD pipelines to speed updates into production. AKS supports the Docker image format, enabling you to automate the deployment of containers. For private storage of your Docker images, you can integrate AKS with Azure Container Registry (ACR), a managed, private Docker registry service based on the open-source Docker Registry 2.0.
- **Azure security**. The migration approaches outlined in this article rely on the underlying Azure infrastructure security, starting with the physical security of the datacenters managed by Microsoft. Sensitive data is protected in all stages. SSL is used to encrypt data coming from the outside—a browser or a device—to the load balancer to the services running in VMs. You can also encrypt direct communications from Natural to Adabas over the TCP/PIP protocol.
- **Better business continuity**. You can bolster application resiliency with a cloud-based disaster recovery solution. For example, you can easily set up a secondary Azure region for disaster recovery and replicate resources from region to another using Azure Site Recovery.
- **Comprehensive, real-time monitoring**. Azure Monitor gives you insight into your applications, infrastructure, and network. It collects monitoring telemetry from a variety of on-premises and Azure sources. You can even use your existing monitoring tool to retrieve telemetry from Azure Monitor or install a monitoring agent on your VMs to communicate health data to your existing monitoring tool.

Next steps

A move to Azure can help your organization quickly add scale and address changing requirements without sacrificing the functionality, security, or compliance that your Adabas & Natural applications provide. With little to no disruption to services you can lift-and-shift your workload to Azure IaaS resources and immediately get the scale, availability, and flexibility of the cloud.

If you take the step of rearchitecting your applications, you can make the most of DevOps processes and toolchains to improve time to market, consolidate your IT landscape, and simplify management.

For more information about Azure components

Azure Availability Azure Availability Zones
What is VPN Gateway?
Understand Azure Load Balancing

Microsoft

Azure mainframe and midrange migration helps organization take advantage of advances in technology and changing economic models by transitioning to cloud services. Azure provides a compelling value proposition with advantages in cost, flexibility, and application maintenance and development.

software AG

Software AG reimagines integration, sparks business transformation, and enables fast innovation on the Internet of Things so you can pioneer differentiating business models. We give you the freedom to connect and integrate any technology from app to edge. We help you free data from silos so it’s shareable, usable, and powerful—enabling you to make the best decisions and unlock entirely new possibilities for growth.

Written by Marlon Johnson, Microsoft Senior Program Manager — Azure Global Engineering with Nanette Ray, Editor, Resources Online. Reviewed by Software AG.