Building secure web applications with Windows virtual machines on Azure

12/06/2018 • 6 minutes to read • Contributors all

In this article

This scenario provides architecture and design guidance for running secure, multi-tier web applications on Microsoft Azure. In this example, an ASP.NET application securely connects to a protected back-end Microsoft SQL Server cluster using virtual machines.

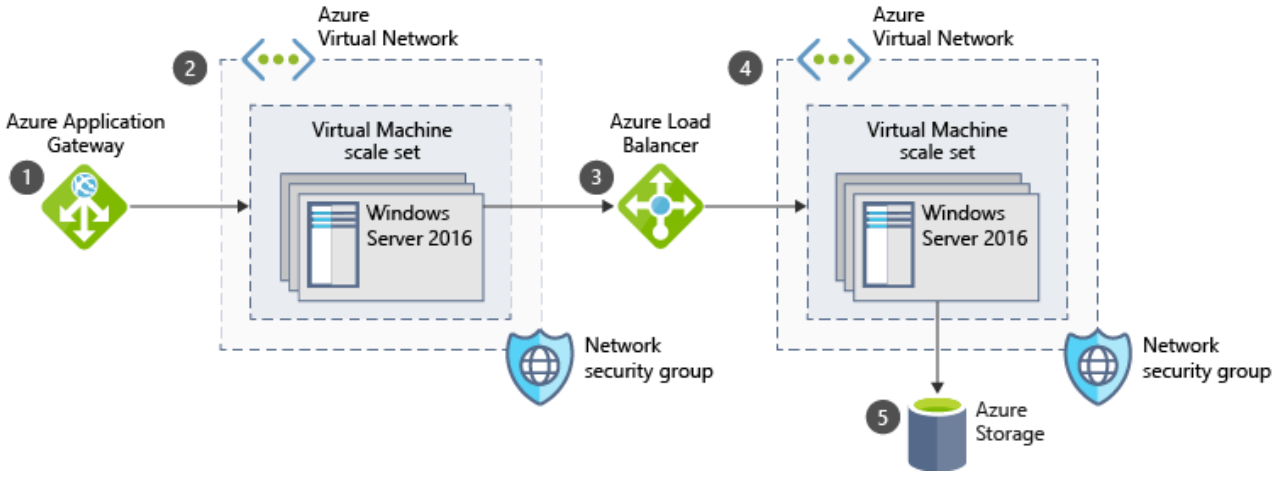
Traditionally, organizations had to maintain legacy on-premises applications and services to provide a secure infrastructure. By deploying these Windows Server applications securely in Azure, organizations can modernize their deployments and reduce their on-premises operating costs and management overhead.

Relevant use cases

A few examples of where this scenario may apply:

- Modernizing application deployments in a secure cloud environment.
- Reducing the management overhead of legacy on-premises applications and services.
- Improving patient healthcare and experience with new application platforms.

Architecture



This scenario shows a front-end web application connecting to a back-end database, both running on Windows Server 2016. The data flows through the scenario as follows:

1. Users access the front-end ASP.NET application through an Azure Application Gateway.
2. The Application Gateway distributes traffic to VM instances within an Azure virtual machine scale set.

3. The application connects to Microsoft SQL Server cluster in a back-end tier via an Azure load balancer. These back-end SQL Server instances are in a separate Azure virtual network, secured by network security group rules that limit traffic flow.
4. The load balancer distributes SQL Server traffic to VM instances in another virtual machine scale set.
5. Azure Blob Storage acts as a cloud witness for the SQL Server cluster in the back-end tier. The connection from within the VNet is enabled with a VNet Service Endpoint for Azure Storage.

Components

- Azure Application Gateway is a layer 7 web traffic load balancer that is application-aware and can distribute traffic based on specific routing rules. App Gateway can also handle SSL offloading for improved web server performance.
- Azure Virtual Network allows resources such as VMs to securely communicate with each other, the Internet, and on-premises networks. Virtual networks provide isolation and segmentation, filter and route traffic, and allow connection between locations. Two virtual networks combined with the appropriate network security groups are used in this scenario to provide a demilitarized zone (DMZ) and isolation of the application components. Virtual network peering connects the two networks together.
- Azure virtual machine scale set lets you create and manager a group of identical, load balanced, VMs. The number of VM instances can automatically increase or decrease in response to demand or a defined schedule. Two separate virtual machine scale sets are used in this scenario - one for the front-end ASP.NET application instances, and one for the back-end SQL Server cluster VM instances. PowerShell desired state configuration (DSC) or the Azure custom script extension can be used to provision the VM instances with the required software and configuration settings.
- Azure network security groups contain a list of security rules that allow or deny inbound or outbound network traffic based on source or destination IP address, port, and protocol. The virtual networks in this scenario are secured with network security group rules that restrict the flow of traffic between the application components.
- Azure load balancer distributes inbound traffic according to rules and health probes. A load balancer provides low latency and high throughput, and scales up to millions of flows for all TCP and UDP applications. An internal load balancer is used in this scenario to distribute traffic from the front-end application tier to the back-end SQL Server cluster.
- Azure Blob Storage acts a Cloud Witness location for the SQL Server cluster. This witness is used for cluster operations and decisions that require an additional vote to decide quorum. Using Cloud Witness removes the need for an additional VM to act as a traditional File Share Witness.

Alternatives

- Linux and Windows can be used interchangeably since the infrastructure isn't dependent on the operating system.
- SQL Server for Linux can replace the back-end data store.
- Cosmos DB is another alternative for the data store.

Considerations

Availability

The VM instances in this scenario are deployed across Availability Zones. Each zone is made up of one or more datacenters equipped with independent power, cooling, and networking. Each enabled region has a minimum of three availability zones. This distribution of VM instances across zones provides high availability to the application tiers.

The database tier can be configured to use Always On availability groups. With this SQL Server configuration, one primary database within a cluster is configured with up to eight secondary databases. If an issue occurs with the

primary database, the cluster fails over to one of the secondary databases, which allows the application to continue to be available. For more information, see Overview of Always On availability groups for SQL Server.

Scalability

This scenario uses virtual machine scale sets for the front-end and back-end components. With scale sets, the number of VM instances that run the front-end application tier can automatically scale in response to customer demand, or based on a defined schedule. For more information, see Overview of autoscale with virtual machine scale sets.

For other scalability topics, see the scalability checklist in the Azure Architecture Center.

Security

All the virtual network traffic into the front-end application tier and protected by network security groups. Rules limit the flow of traffic so that only the front-end application tier VM instances can access the back-end database tier. No outbound Internet traffic is allowed from the database tier. To reduce the attack footprint, no direct remote management ports are open. For more information, see Azure network security groups.

To view guidance on deploying Payment Card Industry Data Security Standards (PCI DSS 3.2) compliant infrastructure. For general guidance on designing secure scenarios, see the Azure Security Documentation.

Resiliency

In combination with the use of Availability Zones and virtual machine scale sets, this scenario uses Azure Application Gateway and load balancer. These two networking components distribute traffic to the connected VM instances, and include health probes that ensure traffic is only distributed to healthy VMs. Two Application Gateway instances are configured in an active-passive configuration, and a zone-redundant load balancer is used. This configuration makes the networking resources and application resilient to issues that would otherwise disrupt traffic and impact end-user access.

For general guidance on designing resilient solutions, see Designing reliable Azure applications.

Deploy the scenario

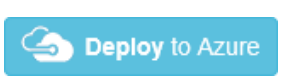
Prerequisites

- You must have an existing Azure account. If you don't have an Azure subscription, create a free account before you begin.
- To deploy a SQL Server cluster into the back-end scale set, you would need a domain in Azure Active Directory (AD) Domain Services.

Deploy the components

To deploy the core infrastructure for this scenario with an Azure Resource Manager template, perform the following steps.

1. Click the link below to deploy the solution.

2. Wait for the template deployment to open in the Azure portal, then complete the following steps:

   - Choose to **Create new** resource group, then provide a name such as *myWindowsscenario* in the text box.
   - Select a region from the **Location** drop-down box.

- Provide a username and secure password for the virtual machine scale set instances.
- Review the terms and conditions, then check **I agree to the terms and conditions stated above**.
- Select the **Purchase** button.

It can take 15-20 minutes for the deployment to complete.

Pricing

To explore the cost of running this scenario, all of the services are pre-configured in the cost calculator. To see how the pricing would change for your particular use case, change the appropriate variables to match your expected traffic.

We have provided three sample cost profiles based on the number of scale set VM instances that run your applications.

- Small: this pricing example correlates to two front-end and two back-end VM instances.
- Medium: this pricing example correlates to 20 front-end and 5 back-end VM instances.
- Large: this pricing example correlates to 100 front-end and 10 back-end VM instances.

Related resources

This scenario used a back-end virtual machine scale set that runs a Microsoft SQL Server cluster. Cosmos DB could also be used as a scalable and secure database tier for the application data. An Azure virtual network service endpoint allows you to secure your critical Azure service resources to only your virtual networks. In this scenario, VNet endpoints allow you to secure traffic between the front-end application tier and Cosmos DB. For more information, see the Azure Cosmos DB overview.

For more detailed implementation guides, review the reference architecture for N-tier applications using SQL Server.