**A 3-tier environment is a common setup. Use a tool of your choosing/familiarity to create these resources on a cloud environment in Azure.**

**Workflow**

* Distribute the VMs in each tier across two availability zones in regions that support zones. In other regions, deploy the VMs in each tier within one availability set.
* The database tier can be configured to use Always On availability groups. With this SQL Server configuration, one primary read/write replica within an availability group is configured with up to eight secondary read-only replicas. If an issue occurs with the primary replica, the availability group fails over primary read/write activity to one of the secondary replicas, allowing the application to remain available. For more information, see [Overview of Always On availability groups for SQL Server](https://learn.microsoft.com/en-us/sql/database-engine/availability-groups/windows/overview-of-always-on-availability-groups-sql-server).
* For disaster recovery scenarios, you can configure SQL Always On asynchronous native replication to the target region used for disaster recovery. You can also configure Azure Site Recovery replication to the target region if the data change rate is within supported limits of Azure Site Recovery.
* Users access the front-end ASP.NET web tier via the traffic manager endpoint.
* The traffic manager redirects traffic to the primary public IP endpoint in the primary source region.
* The public IP redirects the call to one of the web tier VM instances through a public load balancer. All web tier VM instances are in one subnet.
* From the web tier VM, each call is routed to one of the VM instances in the business tier through an internal load balancer for processing. All business tier VMs are in a separate subnet.
* The operation is processed in the business tier and the ASP.NET application connects to Microsoft SQL Server cluster in a back-end tier via an Azure internal load balancer. These back-end SQL Server instances are in a separate subnet.
* The traffic manager's secondary endpoint is configured as the public IP in the target region used for disaster recovery.
* In the event of a primary region disruption, you invoke Azure Site Recovery failover and the application becomes active in the target region.
* The traffic manager endpoint automatically redirects the client traffic to the public IP in the target region.

**Components**

* [Availability sets](https://learn.microsoft.com/en-us/azure/virtual-machines/windows/manage-availability) ensure that the VMs you deploy on Azure are distributed across multiple isolated hardware nodes in a cluster. If a hardware or software failure occurs within Azure, only a subset of your VMs are affected and your entire solution remains available and operational.
* [Availability zones](https://learn.microsoft.com/en-us/azure/availability-zones/az-overview) protect your applications and data from datacenter failures. Availability zones are separate physical locations within an Azure region. Each zone consists of one or more datacenters equipped with independent power, cooling, and networking.
* [Azure Site Recovery](https://learn.microsoft.com/en-us/azure/site-recovery/azure-to-azure-quickstart) allows you to replicate VMs to another Azure region for business continuity and disaster recovery needs. You can conduct periodic disaster recovery drills to ensure you meet the compliance needs. The VM will be replicated with the specified settings to the selected region so that you can recover your applications in the event of outages in the source region.
* [Azure Traffic Manager](https://learn.microsoft.com/en-us/azure/traffic-manager) is a DNS-based traffic load balancer that distributes traffic optimally to services across global Azure regions while providing high availability and responsiveness.
* [Azure Load Balancer](https://learn.microsoft.com/en-us/azure/load-balancer/load-balancer-overview) distributes inbound traffic according to defined rules and health probes. A load balancer provides low latency and high throughput, scaling up to millions of flows for all TCP and UDP applications. A public load balancer is used in this scenario to distribute incoming client traffic to the web tier. An internal load balancer is used in this scenario to distribute traffic from the business tier to the back-end SQL Server cluster.