7- Private endpoints

Some Azure resources, such as Azure SQL Database instances and Azure Storage containers, can't be put in a virtual network directly, but there is an indirect way to bring them into a VNet. The solution is called a private endpoint. It's simply a private IP address in a virtual network that's connected to an Azure resource that's outside of the VNet.

If you don't use a private endpoint, then the resources in your VNet have to connect to the external resource using a public endpoint. For example, a virtual machine in your VNet would access an Azure Storage container using the container's public endpoint. The problem with public endpoints is that they're exposed to the internet, which is a security risk. If you created a private endpoint for your Azure Storage container, you could disable its public endpoint. Then the only way to access your storage container would be to connect to it over the Microsoft backbone network from your VNet.

*I should mention that not all Azure resources can be connected to a private endpoint. A resource has to be hosted by a service that supports something called **Private Link**. This is what's actually used to connect your private endpoint to the service.*

Here's how all of this works. Suppose you have a virtual machine that's running an application that needs to store its data in a SQL Database instance called DB1. Let's say the VM is in a subnet called Sub1 in a virtual network called Vnet1. First, you'd create a private endpoint called, let's say, PrivateSqlEndpoint. You'd configure it to be in the Sub1 subnet in Vnet1, and you'd configure its target to be DB1. Next, you'd need to go into DB1's configuration and set its connectivity method to private endpoint. You could also disable public access to DB1 so the database would only be accessible through a private endpoint.

Once this was set up, the application on the VM could access DB1 by connecting to the IP address of PrivateSqlEndpoint. Behind the scenes, the Private Link service would send the traffic over **Microsoft's backbone network** from the private endpoint to DB1. This is such a handy solution that Microsoft has even made it possible to set up a custom Private Link service for your own application.

Private endpoints are also a great solution when you have other networks connected to a VNet. For example, if your VNet is peered to another VNet, then resources in the peered VNet can access an external Azure resource through a private endpoint as well. This even works when you have an on-premises environment connected to your VNet.