**README**

This terraform script deploys an Azure API Management with internal integration to a Virtual network in a new resource group.

It also creates an app service plan with App Slots for Staging and Production.

Azure API Management is a hybrid, multicloud management platform for APIs across all environments. As a platform-as-a-service, API Management supports the complete API lifecycle.

APIs simplify application integration and make data and services reusable and universally accessible. With the proliferation and increasing dependency on APIs, organizations need to manage them as first-class assets throughout their lifecycle.

Azure API Management helps customers meet these challenges:

* Abstract backend architecture diversity and complexity from API consumers
* Securely expose services hosted on and outside of Azure as APIs
* Protect, accelerate, and observe APIs
* Enable API discovery and consumption by internal and external users

For more on API Management, see <https://learn.microsoft.com/en-us/azure/api-management/api-management-key-concepts>

The version of AzureRM used in my project is 2.1.0 which is the latest at the moment. The "azurerm\_api\_management" resource had some issues with earlier versions.

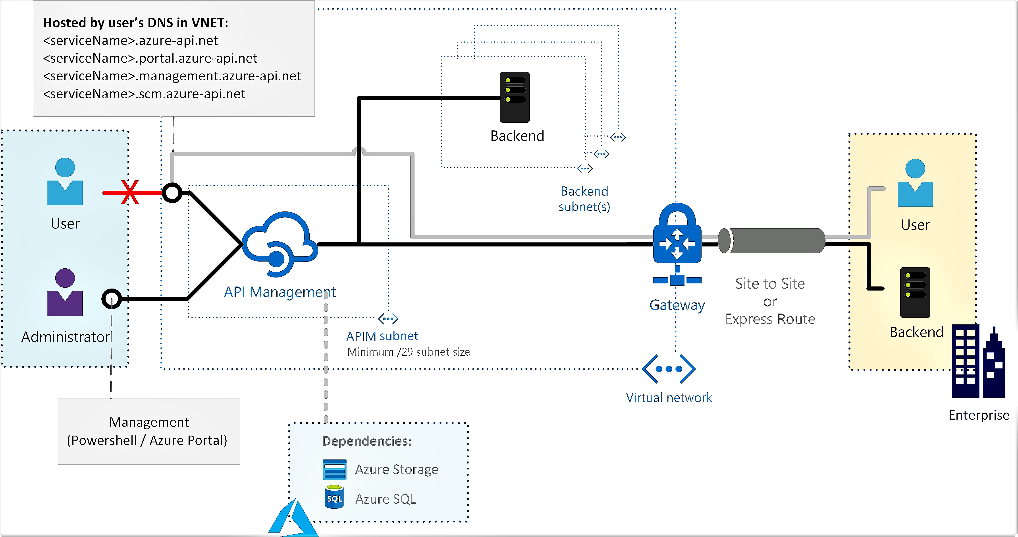
First we create the resource group where everything will end up in:

Then we create the virtual network that the API Management service will be integrated with.  
With Azure virtual networks (VNets), Azure API Management can manage internet-inaccessible APIs using several VPN technologies to make the connection.  
In this mode, you can only access the following API Management endpoints within a VNet whose access you control.

* The API gateway
* The developer portal
* Direct management
* Git

Use API Management in internal mode to:

* Make APIs hosted in your private datacenter securely accessible by third parties outside of it by using Azure VPN connections or Azure ExpressRoute.
* Enable hybrid cloud scenarios by exposing your cloud-based APIs and on-premises APIs through a common gateway.
* Manage your APIs hosted in multiple geographic locations, using a single gateway endpoint.



For configurations specific to the external mode, where the API Management endpoints are accessible from the public internet, and backend services are located in the network, see [Connect to a virtual network using Azure API Management](https://learn.microsoft.com/en-us/azure/api-management/api-management-using-with-vnet).

Next step is the core resource, APIM itself.

We create a resource in the developer tier with a certain capacity as vnet integration is only available in Developer and Premium tier. Most of the settings are correct by default, like enabled TLS version, so we don't need to specify a lot.

If you use policies, you can store them as a separate file in a Storage Account, then you need to specify a publicly accessible URL but the policies are not considered here.

Finally is the app service plan with for staging and production environments.  
When you deploy your web app, web app on Linux, mobile back end, or API app to Azure App Service, you can use a separate deployment slot instead of the default production slot when you're running in the **Standard, Premium**, or **Isolated** App Service plan tier. Deployment slots are live apps with their own host names. App content and configurations elements can be swapped between two deployment slots, including the production slot.