I will show you a generic template, in this case a VNet with some additional resources, whereby nearly nothing is hard coded and is entirely driven by the parameter file.

Open the template Demo1.Generic.json

Start by showing the VNet parameter is an Object. Then scroll to the VNet resource to show how nothing is hard coded and that all properties are referencing the Object for a value. Further, show how flexibility (thanks to the Parameter file) can be incorporated by making some child/nested resources be based on the ‘copy’ property. In this case, the ‘subnets’ property is actually a copy loop and will loop based on the ‘subnets’ array count from the parameter file.

Switch over to the Demo1.Generic.parameters.json parameter file. Scroll through slowly to show all the various properties. A key area to highlight is ‘Subnets’ which has “3” entries in its array. State how that means the Copy loop in the Template file will look 3 times

Deploy the template

Further, show the generic templates can be better integrated into Enterprise given how the values can be more manipulated as code. To show this:

* Import the Demo1.Generic.Parameters.json file into PowerShell
  + $Parameters = Get-Content .\Demo1.Generic.parameters.json | ConvertFrom-Json
  + Show each (in order)
    - $Parameters
    - $Parameters.Parameters
      * (Highlight how it matches our original 3 parameter values, feel free to switch back to the parameter json file)
    - $Parameters.parameters.VNet.value
    - $Parameters.parameters.VNet.value.Subnets
    - $Parameters.parameters.VNet.value.Subnets[0].AddressPrefix = "10.2.0.0/24"
    - $Parameters.parameters.VNet.value.Subnets
      * Show the results of changing the parameter file to reflect a new address prefix
    - Incorporate the parameter changes into a NEW parameter file “Demo1.Generic.Parameters2.json”
      * $Parameters | ConvertTo-Json -Depth 20 | Out-File Demo1.Generic.Parameter2.json -Verbose
* The new parameter file, Demo1.Generic.Parameter2.json will look a bit bulkier due to the formatting but the ARM instructions are still the same. The formatting can be easily cleaned up and looking normal by using any readily available JSON Lint web tool on the Internet.
* Deploy the template with the new parameter file Demo1.Generic.Parameter2.json to show it deploys successfully
* Explain that this method of modifying parameter files opens the door to automation. For example, imagine grabbing the needed VNet address space by programmatically pulling it from an enterprise IP management solution (IPAM)