(Lab guide consideration – will the resources to be deployed take too long for each deployment and what’s the likelihood the audience will ‘connect’ with what’s being deployed)

Goal: Show how the same template can be deployed multiple ways. Show flexibility of template reuse by changing parameters with each deployment

Template details:

CopyIndex is not a learned topic till the next module

Idea 1: 1 Vnet – 2 subs, Avail Set, VM, 2 NICs, 1 OS, 1 Data disk, Storage acct

Idea 2: App Svc / Web App, Storage Acct

All resources should be deployed to same RG that way their progress can be viewed for each deployment (**Issue** – will concurrent deployments of /different resource names/ to the same RG cause a deployment failure?)

1. Portal - (Quickly) deploy via Portal’s Template Deployment – load .json, fill out parameters, deploy
2. PS – Files – Deploy template file & parameter file, deploy
3. PS – in-line – Deploy template file & parameters are in-line as arguments
4. PS – URI – Deploy template URI & parameter URI
   1. **Issue** – Content must be available publicly, possibly an ASD hosted GitHub

Linked / Nested Templates

Explain that decoupling ‘environment planes’ provides flexibility from the perspective of ‘a person’s role authority’ or simply for ease of deployment administration/upkeep. Example: Why modify the “network” resources written into every single use-case template when you could instead just have templates dedicated to “network” resources only

Idea 1: Master template – (doesn’t seem as ‘real life’ as it could be) – pre-create RGs or deploy “subscription scope” template to create RGs (not covered in slide decks yet)

* Root template (RG-A) will be purposed to deploy template history only
* (RG-B) Linked template will create a “vnet” – a parameter determines whether it’ll hit the Production Vnet json or the Development Vnet json. Difference between them may be the # of subnets
* (RG-C) Nested template will create a log analytics workspace and/or Azure DNS Zone & “mail” record pointing to fictitious address

1. Deploy template. Show how the child deployment resources appear in the ‘deployed to’ RG in that no actual resources are deployed to RG-A but it maintains the deployment history. Show the resources in the other RG
2. Deploy again showing only a single change in a parameter of the Master template will now go to the other linked template. By doing it this method “departments”, like the Network team can maintain their templates as needed without affecting the work of “other departments” templates.