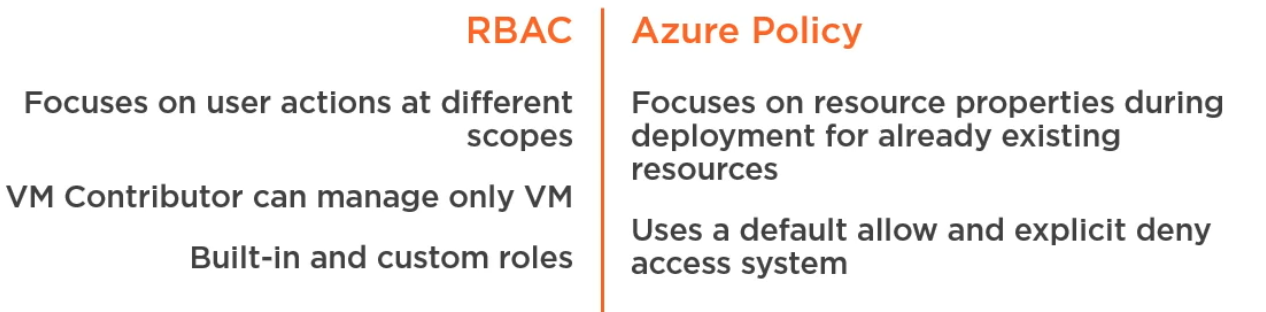
RBAC: Role Based Access Control: governs what an authenticated and authorized user can or cannot do.

Azure Policy: Focuses on deployments (allowed configs, etc)



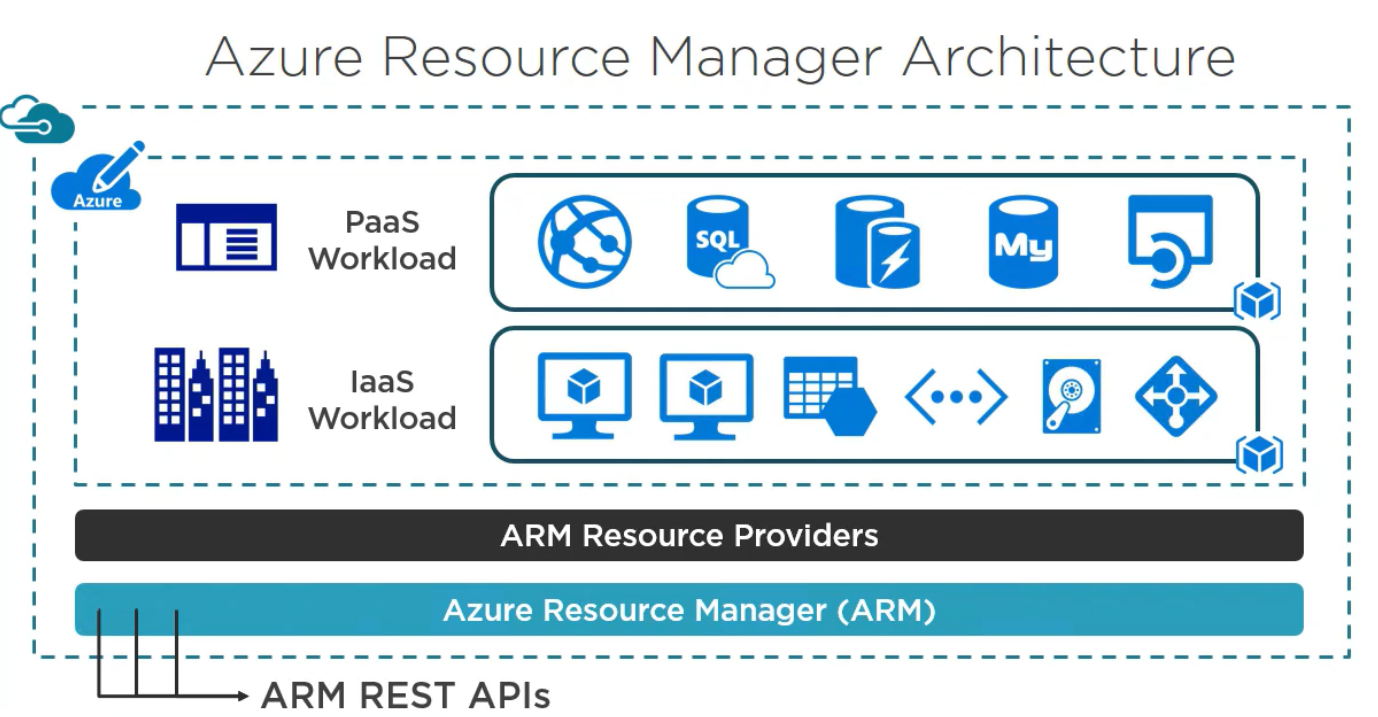
Some default Azure Policies:

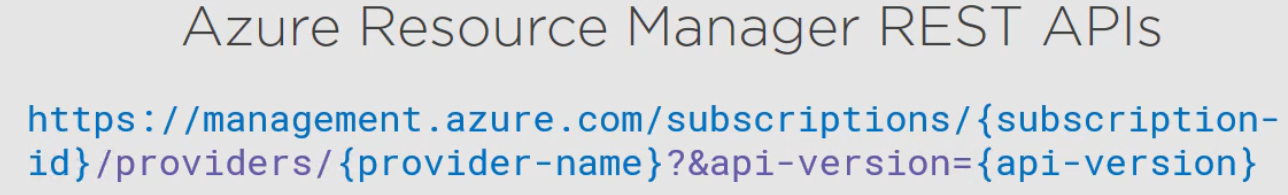
* Allowed Locations
* VM SKUs (Stuck Keeping Units): controlling sizes of VMs
* Deploy MMS extensions (extension used to onboard VM into Azure Log Analytics) ~OMS

**Definitions:** Collections of policies

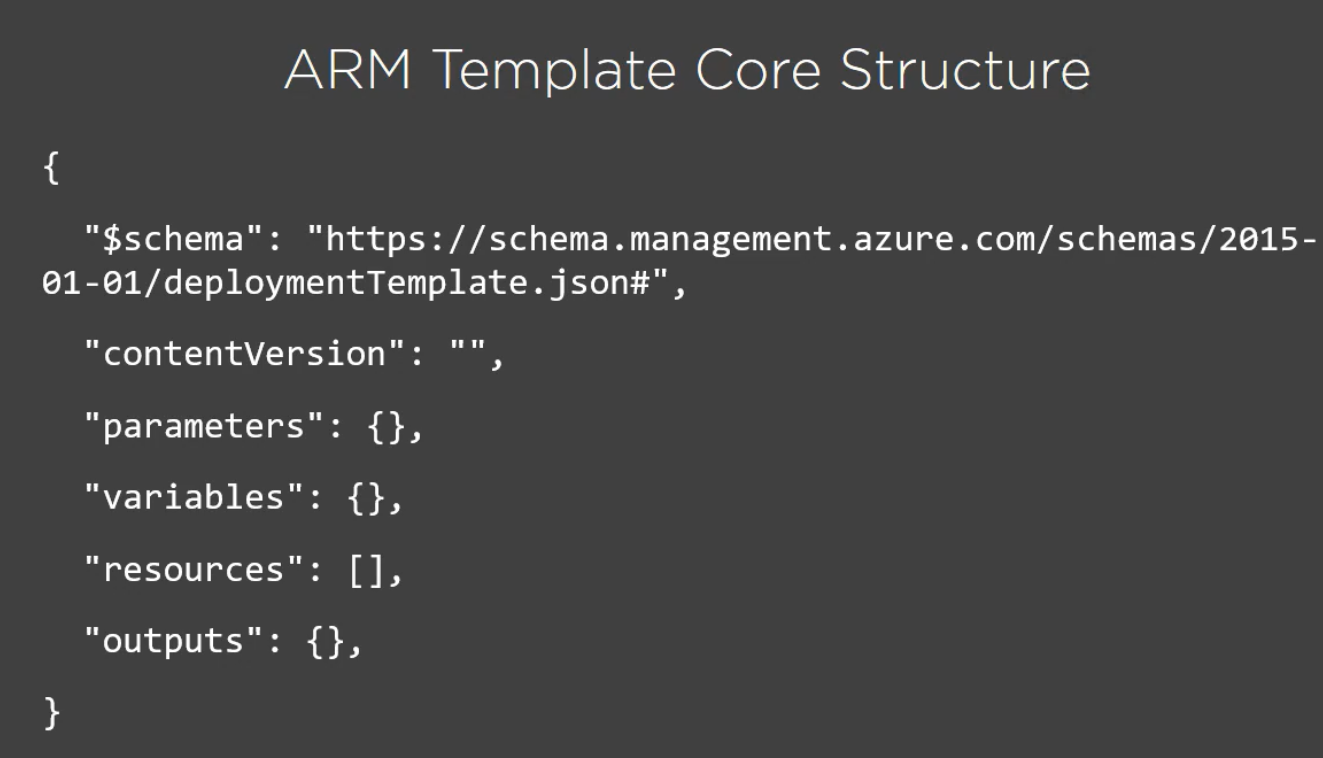
-Mode Incremental/Complete: incremental does not delete any other item in the resource group. Complete deletes everything before running the template



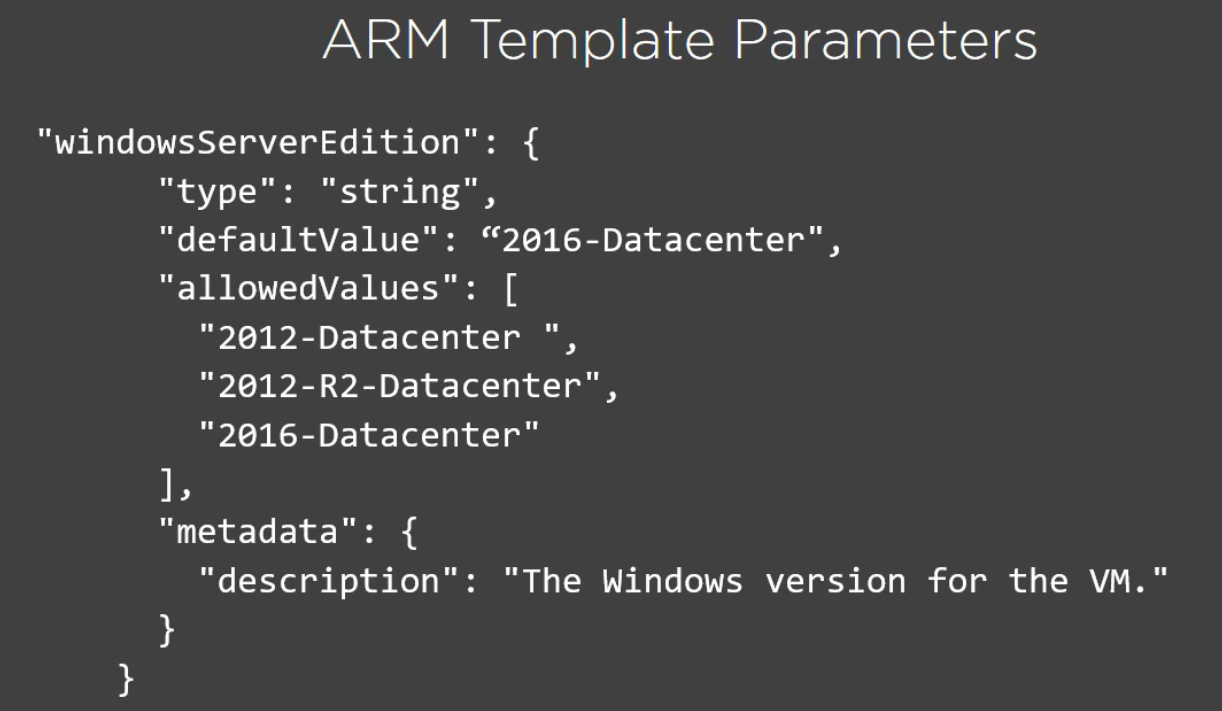




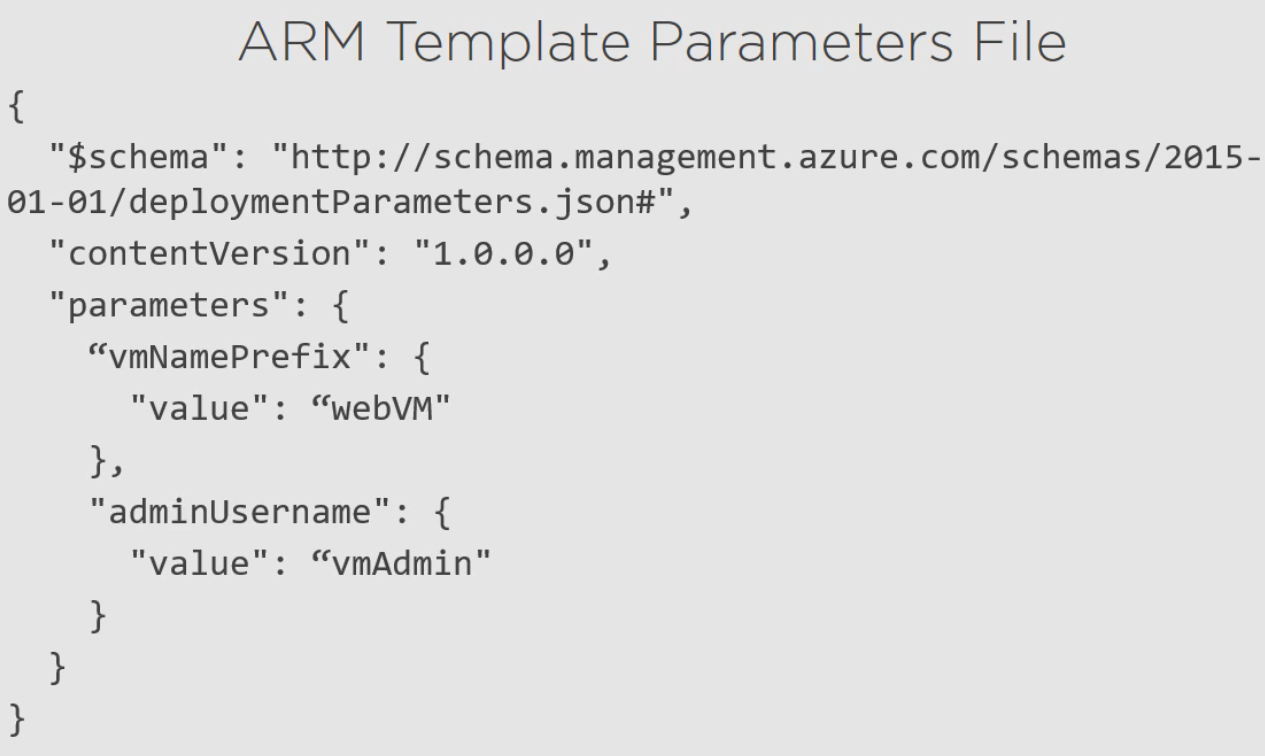
* Resource grouping
* Resource dependencies (define deployment order, etc)
* Repeatable deployments
* Deployment templates
* RBAC (role based access control)
* Granular billing, resource tagging



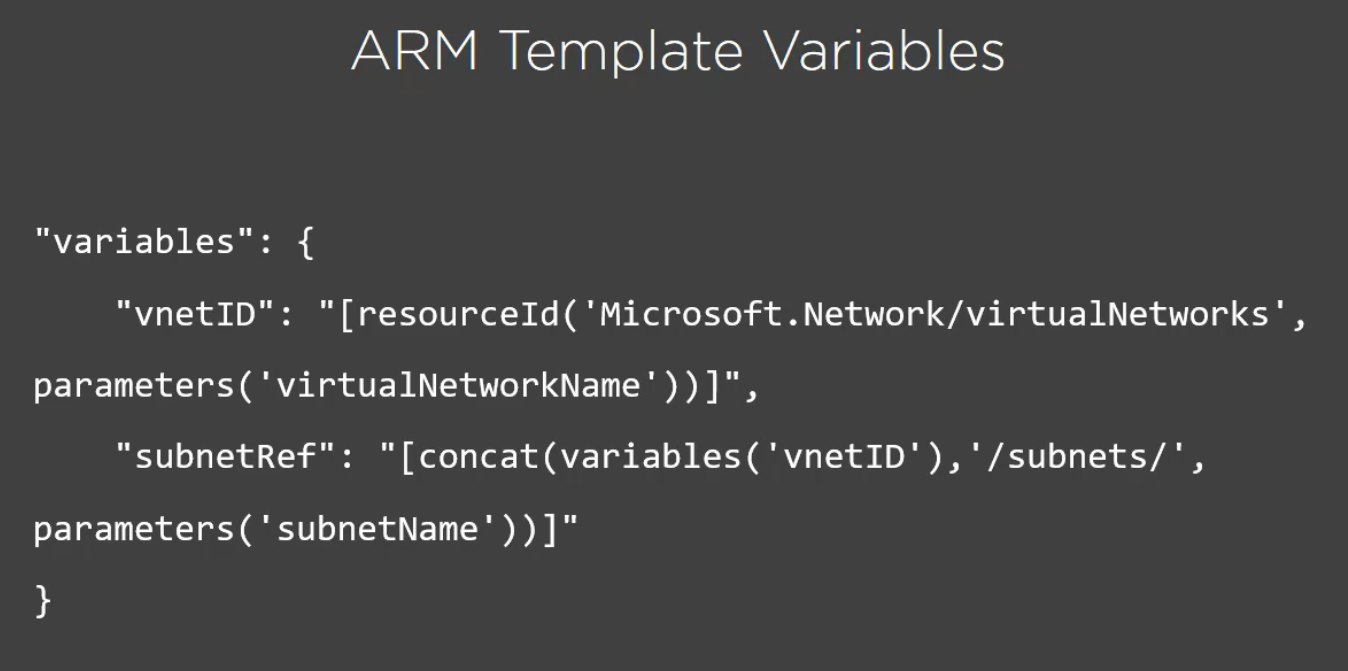
* **schema**: how to validate structure of the template (+ VS provides additional intellisense n stuff like that depending on this during development)
* **contentVersion:** version number, metadata for the developer, point of reference
* **parameters**: provided by the developer (eg. VM's name)
* **variables**: ARM can calculate these on the fly
* **resources**: array
* **outputs**: optional, eg. webapp's URL can be printed here

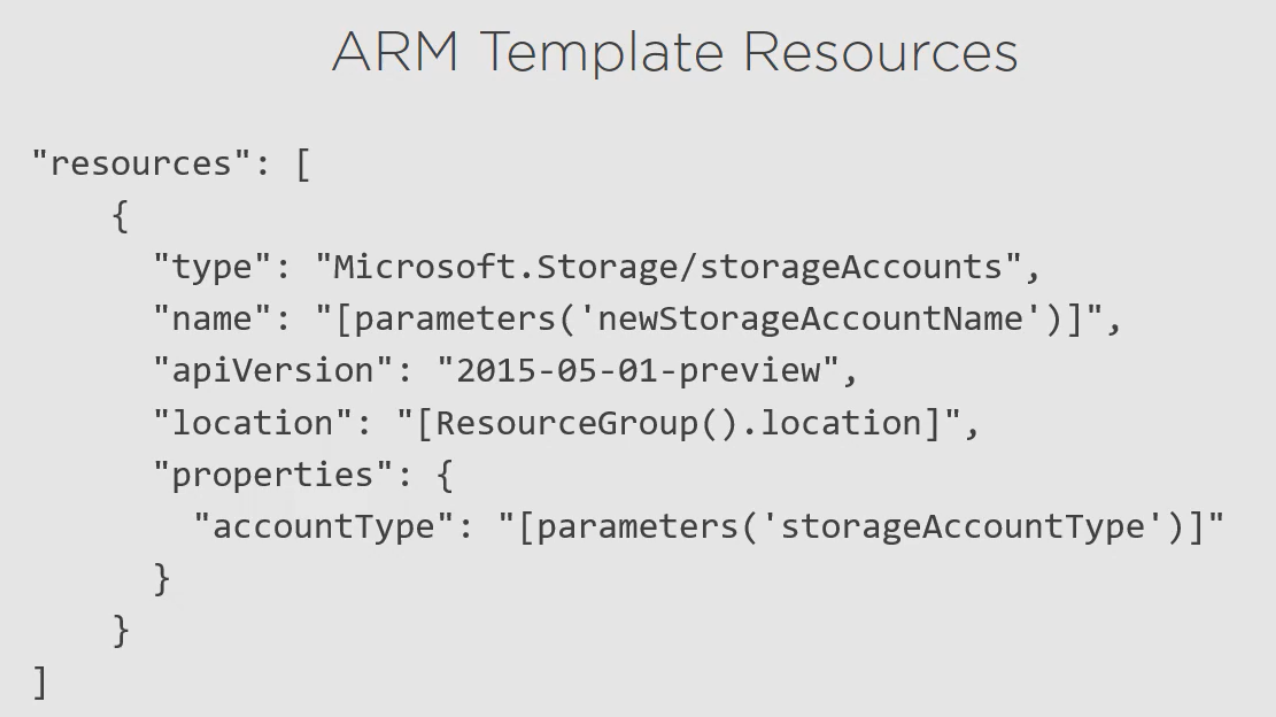


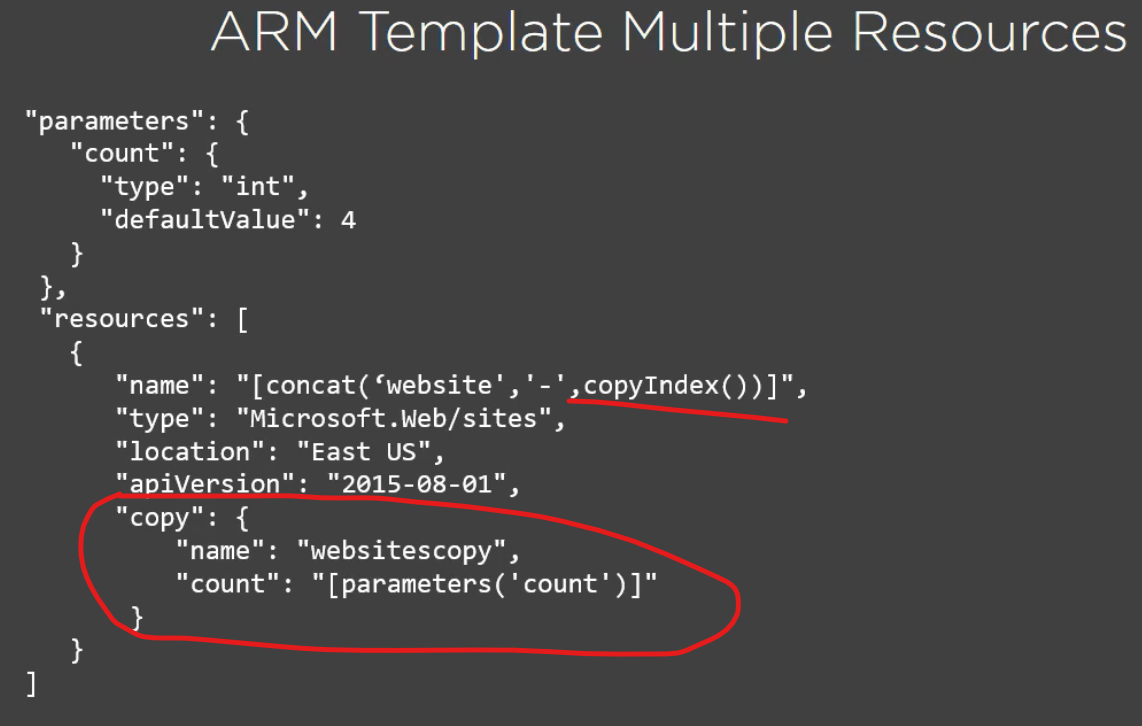
* only mandatory property in their JSON is: **type** (eg. a string, or array, int, securestring, etc)

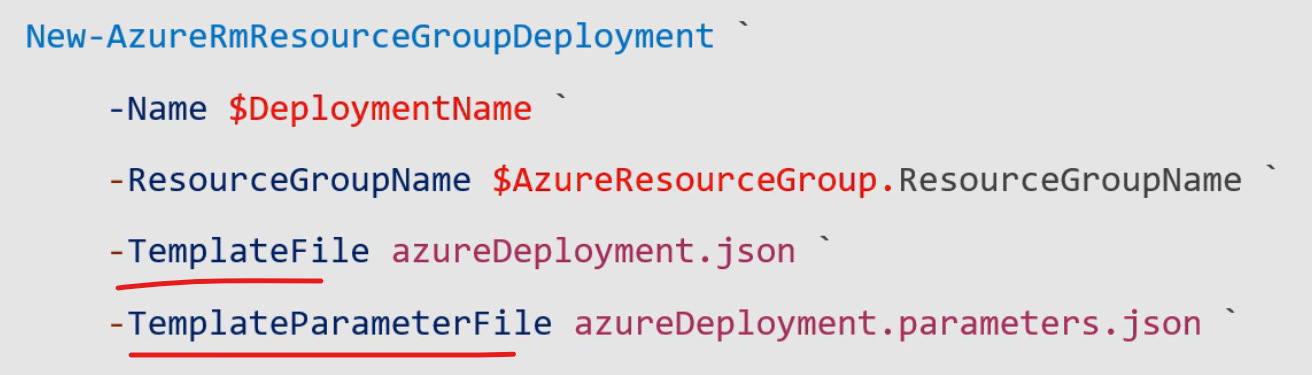


This is all plain text, cannot be encrypted

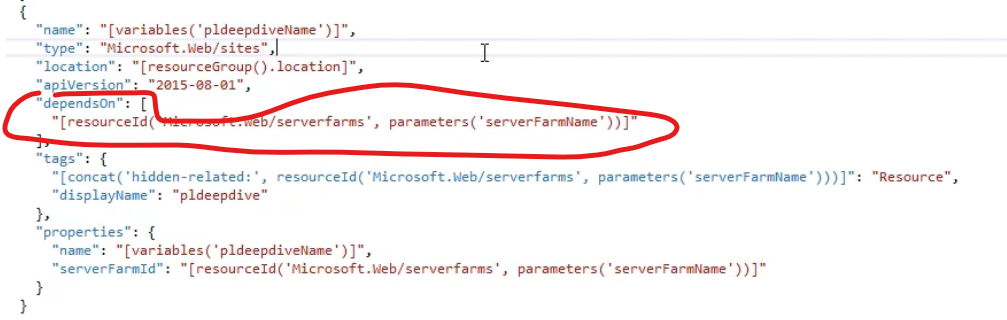








Highlighted ones can be URLs (or files from like source control)  
+ Additional hashtable of additional parameters (anything really 🡪 can be good if we want to store some parameters, but not in a separate JSON)



Order of stuff in the template JSON does not matter: parallel builds are used when Azure is creating the resources from the JSON 🡪 just make sure the **dependsOn** fields are set correctly