I will demonstrate how to insert new resource types that otherwise aren’t available as user snippets or the user snippets aren’t using the latest features & API.

Incorporating resource types from scratch may sound like it takes awhile, but it can actually be faster than using existing templates from the Github Azure quickstart template gallery. Using existing templates from the gallery is fine for proof of concepts or learning but modifying them (and troubleshooting) could take longer considering it needs to be adopted to your organization’s strategies.

It’s important to note that not every property you find on this page is editable by you. Some properties are exclusively used by the Azure Fabric even though they are still listed on the Azure Template Reference page (since the page is basically a Schema dump).

You will likely NOT use all available and editable properties to you either because they don’t fit your needs or to simply reduce the overall complexity

* Let’s start by creating a new JSON file in VS Code
* The only user snippet I’ll be incorporating is the generic template skeleton framework
  + “arm!”
* Next I’ll navigate to the Azure Template Reference page. This page lists the various resource providers, the API versions they support, the resource types available from that specific API, and a listing of all possible properties for that resource type
* I’ll go ahead and incorporate a few common resource types and select newer resource types
* Storage Account : (Select the following – Storage -> 2019-04-01 (or latest) -> Storage Accounts
  + Scroll through the page slowly and show how all the properties have generic values that indicate the type of input they require
  + Next scroll to the section “Property Values” to showcase how each property in the hierarchy of that resource type is listed including its purpose and possibly what exact values it accepts. If any of the property names look new to you, it is because PowerShell and the Azure Portal may not necessarily expose all possible properties to you.
  + Scroll back to the top of the page and click the “Copy” button to copy the contents of the Storage resource
  + Return to VS Code and paste the contents on a new line within the Resources [ ] section
    - A lot of the code sections can be removed since we simply do not want to use it. Keyword is want. Remember, not every editable property should necessary be included.
  + Remove sections you don’t plan to use.
    - Show 2 ways to remove it
      * 1. Highlight the “Restrictions” section between and including the square brackets [ ]
      * 2. Navigate to the “Network ACLs” section. Collapse its tree by clicking the Dashed line or down carrot symbol which will collapse all text between the brackets [ ]. You can then just highlight the [ ] for ensuring you selected and remove only section
      * 3. Remove the other sections: Encryption, CustomDomain, azureFilesIdentityBasedAuthentication, isHnsEnabled, identity, supportsHttpsTrafficOnly
      * Let’s start adding in property values and make use of intellisense!
        + Remove each property’s default value like “string” and use intellisense when necessary to provide possible values. If intellisense doesn’t provide what you need, you can always try building the resource type in the Azure Portal and use the ‘automation template’ or ‘export template’ buttons to see an example of a finished product as code
  + If you get stuck or want to see a finished product that incorporated minimal code, see Demo2\_Example.json