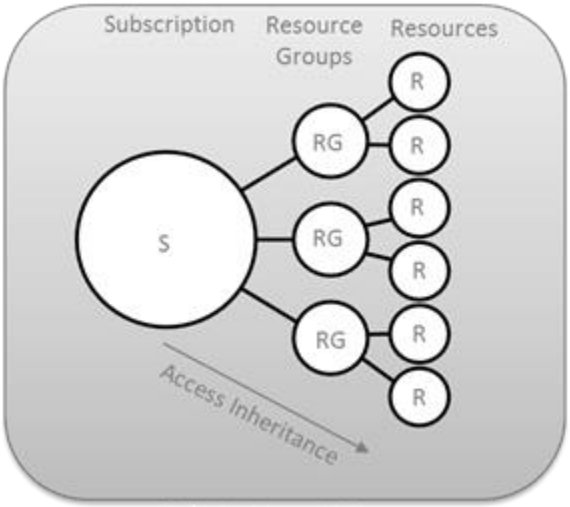
Getting Started (Resources & Services)

Now we've got user stuff more or less squared away (or at least have something shaky to stand on…), let's get into the fun stuff… resources and services.

A service is pretty self-explanatory. A resource is sort of like a service, but it could be other things that support a service as well (a server instance for a SQL Database, or an App Service Plan that covers a few Web Apps, etc.). Let's consider a resource to be any actual thing that we're going to work with, *except* Active Directory and Subscriptions.

Resource groups are containers for resources that *aide in managing resources as a single unit*. I emphasize this because they are only an aide and it is not required that two resources be in the same resource group to work together. So, it's perfectly "legal" to have a SQL Database instance in one resource group and a Web App that uses the database in a separate resource group. Note that resource groups cannot be shared by subscriptions (but, a given user might be able to access various subscriptions' resources/resource groups, if they're given the appropriate permissions through Active Directory: again the Subscription/AD/Services relationships making things a little muddy).



Role Assignment Scopes

Resource groups provide two advantages:

1) They allow you to monitor (and manage) resource usage as a single unit, so you can adjust various parameters for all contained resources at once, and
2) They help with the general organization of things. If you have two unrelated websites that you'll support, each of their services (resources) can go into its own resource group, and then you can management them separately and have things remain fairly clean in the portal.

Resource groups are a pain in the ass… at the time of writing anyway, because…

- The only way to move a resource from one group to another is via PowerShell, and even that's questionable. MS has a lot of strong feedback about being able to move resources from one group to another via the portal UI, and are supposedly in process of working on it (they've since released this PS means, but judging by current findings, it's hit or miss). So for practical purposes, let's assume that a resource cannot be moved from one group to another.
- Next, consider this: by default, every time you create a new resource, Microsoft creates a **new** resource group for it! Talk about a mess… good thing you're learning this before you go make a bunch of databases and websites huh?
- Some resources have to be "tricked" into being made as part of an existing resource group. For example, you can't create a SQL Server instance directly, but only through the process of creating a new SQL Database, so if you want a server named mycompany.database.windows.net that will sit in the MyCompanyHybridApp resource group, then you need to first create the resource group, then you need to start through the process of creating the database, and in that process you'll be prompted to use a new server or an existing one. You'll give the new server a name, you'll specify the resource group for the database, and only then will you have all semantically related resources under the same resource group.

That's about all I'm going to cover for resources and resource groups at the moment. Basically just wanted to get the concept out of the way before we start blindly provisioning databases and web apps because it's a hell of a task to get them in line after the fact.

June 22nd, 2015
Jack D. Leach
Dymeng SERVICES
http://www.dymeng.com/azure