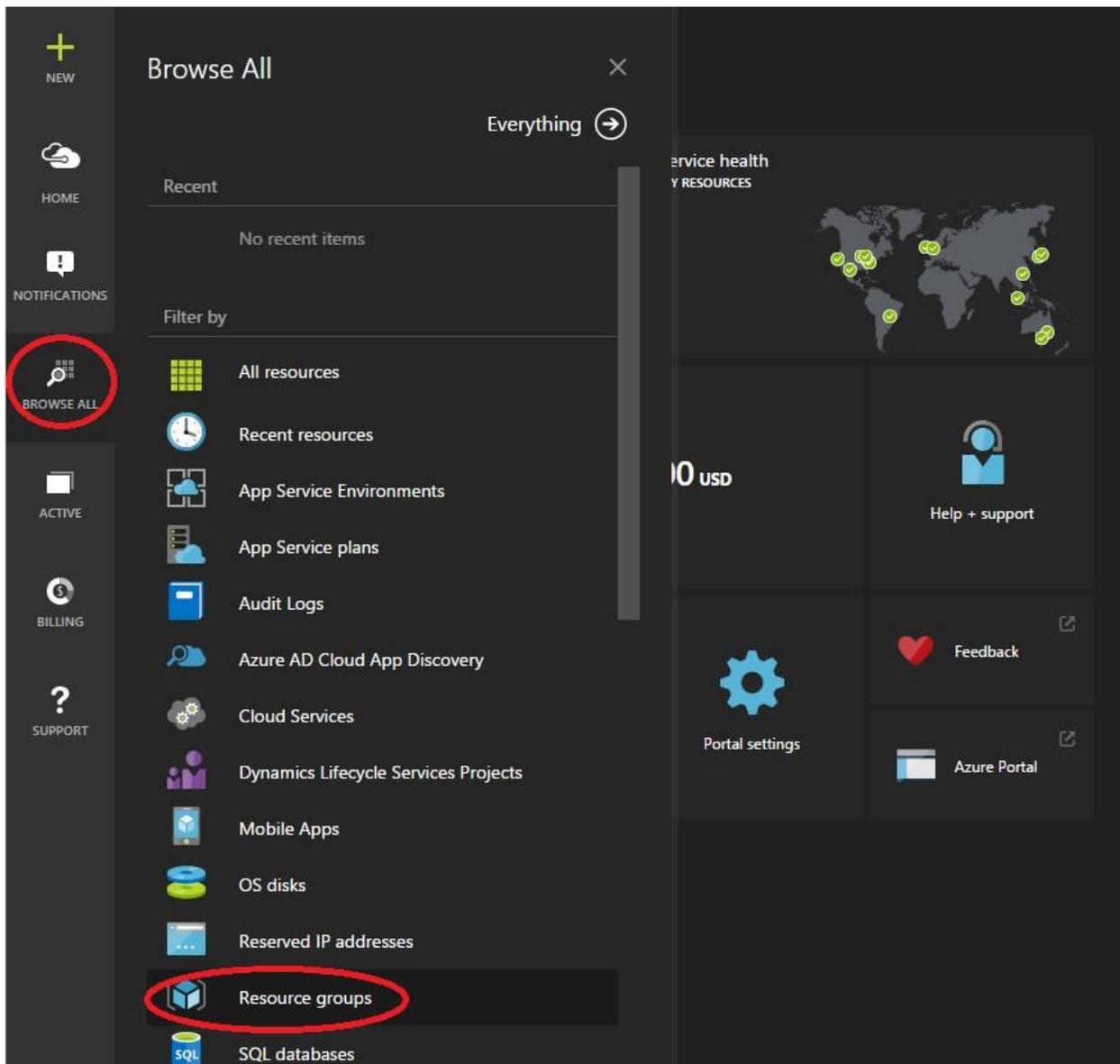


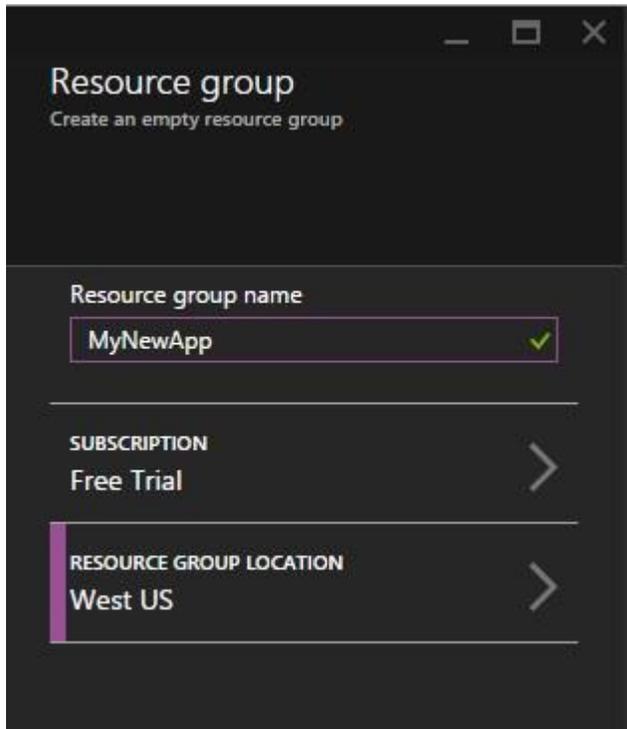
## SQL Database

(Formerly called SQL Azure, the official name for this service is specifically “SQL Database”). We know a bit about users, accounts and resource groups, so let’s get our database up and rolling. Open up the portal (<https://portal.azure.com>) and click the Browse All icon on the left side jumpbar. This opens a blade (e.g., slide out window that MS has dubbed a Blade), scroll through the list until you see Resource Groups (remember from the last article that we want to create a resource group first, then create the SQL Database within that resource group).



(I don’t like the default blue at all. You can change the color by going to Settings... click your name in the top right of the Azure portal and look for Settings in the dropdown list).

This opens a larger Resource Groups blade with an empty list and a top bar (not sure what that one's called...), with an Add, Choose Columns and Refresh icon/button. Click the Add icon, of course, which opens yet another blade.

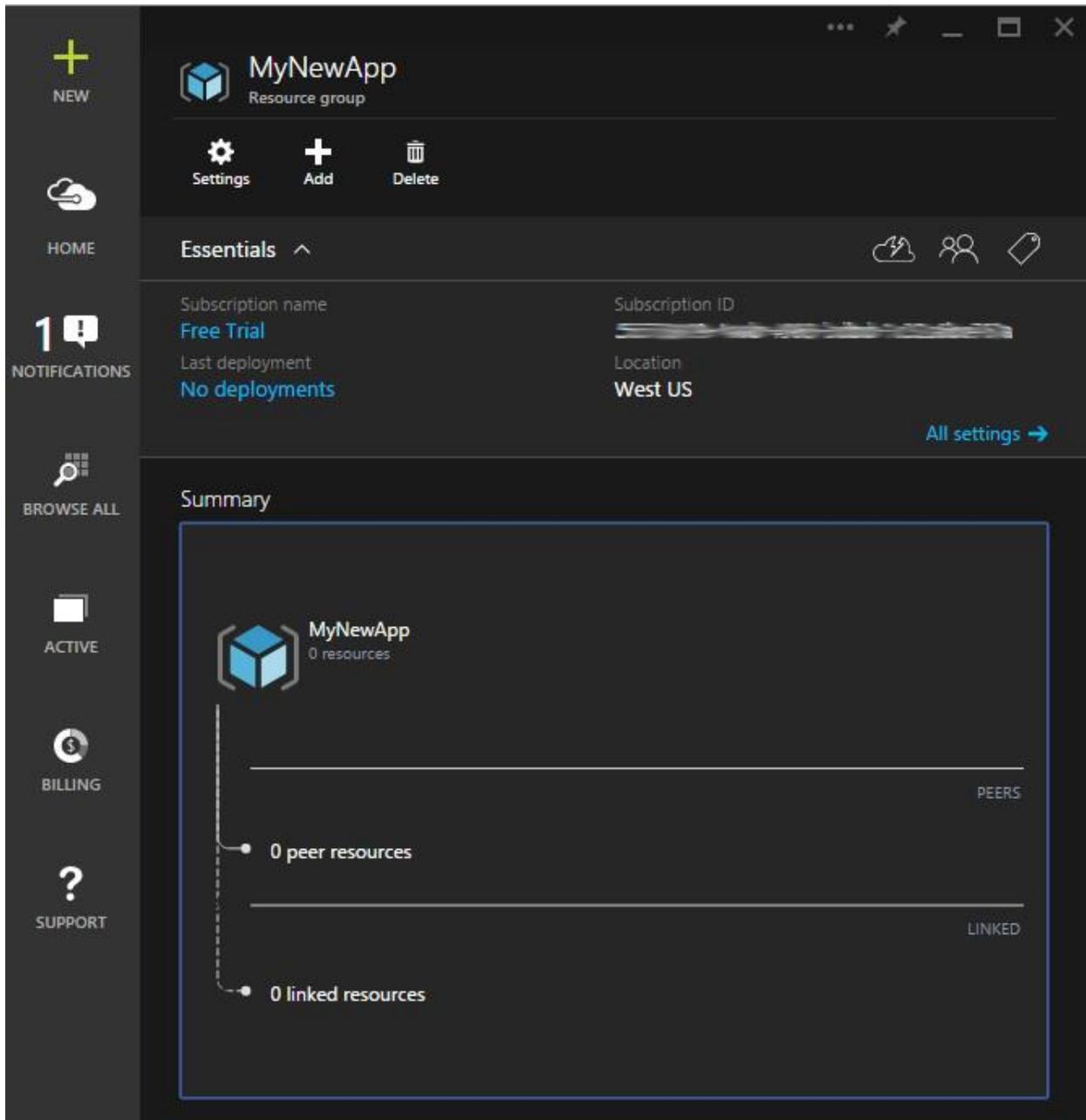


Give the resource group an name that resembles whatever project you intend. Choose a subscription that's available, and choose a resource group location (datacenter) that will act as a template for resources created within the group (don't worry, you can include resources from any data center into a resource group regardless of the resource group's location... in fact, I'm not sure that this setting really does anything practical, but nevertheless it is required).

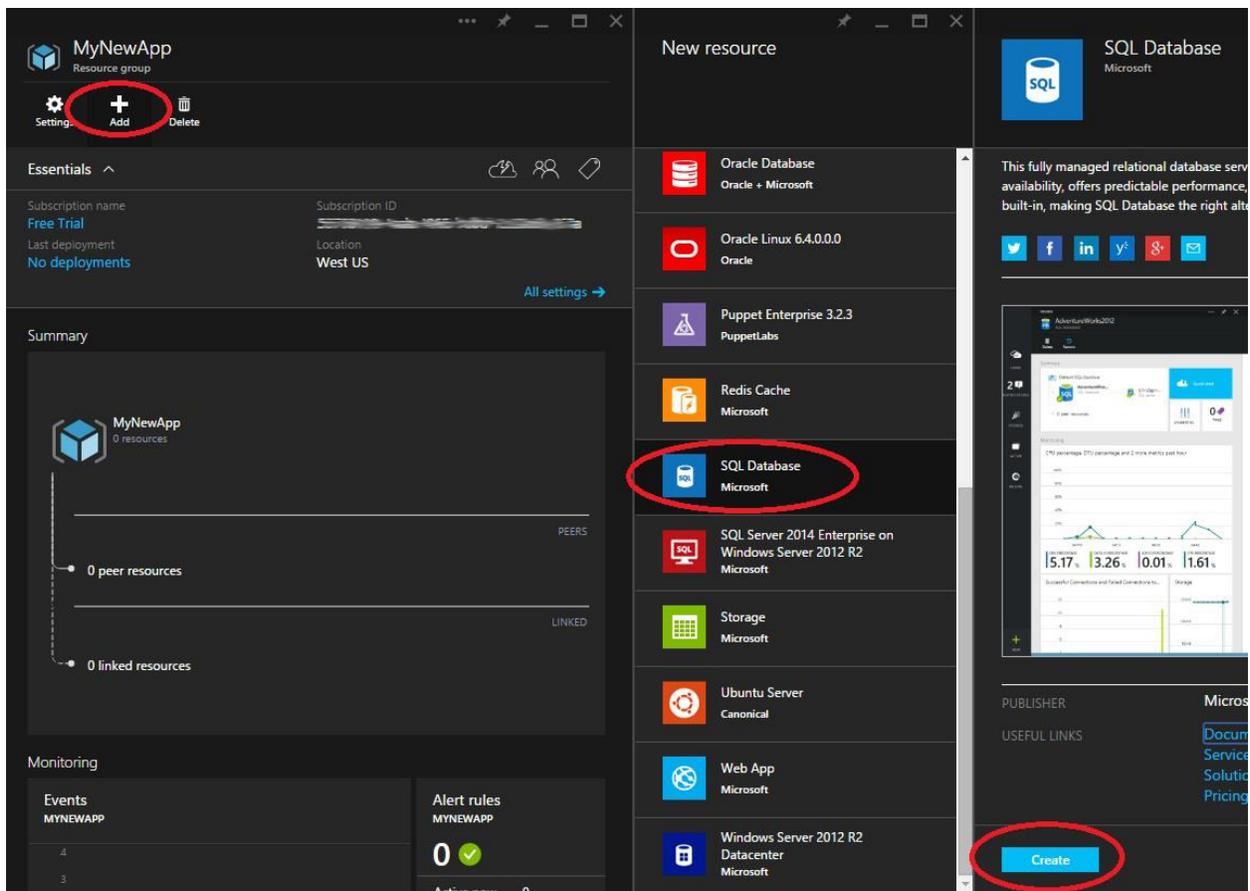
Once those are done click Create at the bottom (and optionally pin it to your home page, which I think is a decent idea as it's quick access to the rest of the stuff we'll have).

Give it a second to provision the resource group and you'll then see this:



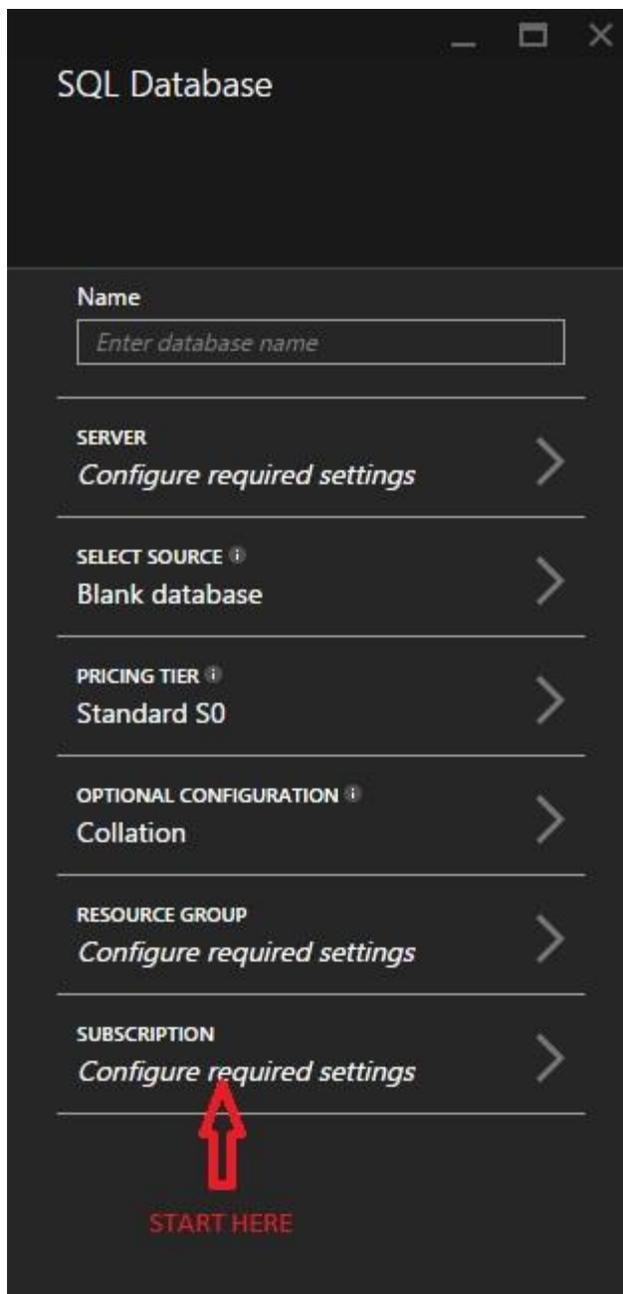


Congratulations. Now let's get to our database... in this blade, click the Add button at the top that's under new your new resource group's name. This will open another list blade: scroll down and select SQL Database from that list. In the resultant blade, click Create at the bottom to start setting it up:



Ok, **STOP** here!

Hopefully this will be fixed by the time someone gets around to reading this, but we'll be on our new SQL Database settings blade, and you'll see a list of things that need configuration. *If you wait until last to select the Subscription, as seems reasonable given that it's the last in the list... the Subscription settings will oh so conveniently lock for you, so you can't select anything and have to exit the whole thing. So, work from the bottom up: setup the Subscription first, then the resource group, then everything else...*



Select a subscription. Straightforward enough, you probably only have one. This selects a default resource group for you, but not the one you want. Click that setting and select the resource group you previously created (lest a new, entirely unrelated resource group be created for you). Set your collation (currently defaulting to SQL\_Latin1\_General\_CP\_CI\_AS, but I wouldn't hold my breath on that).

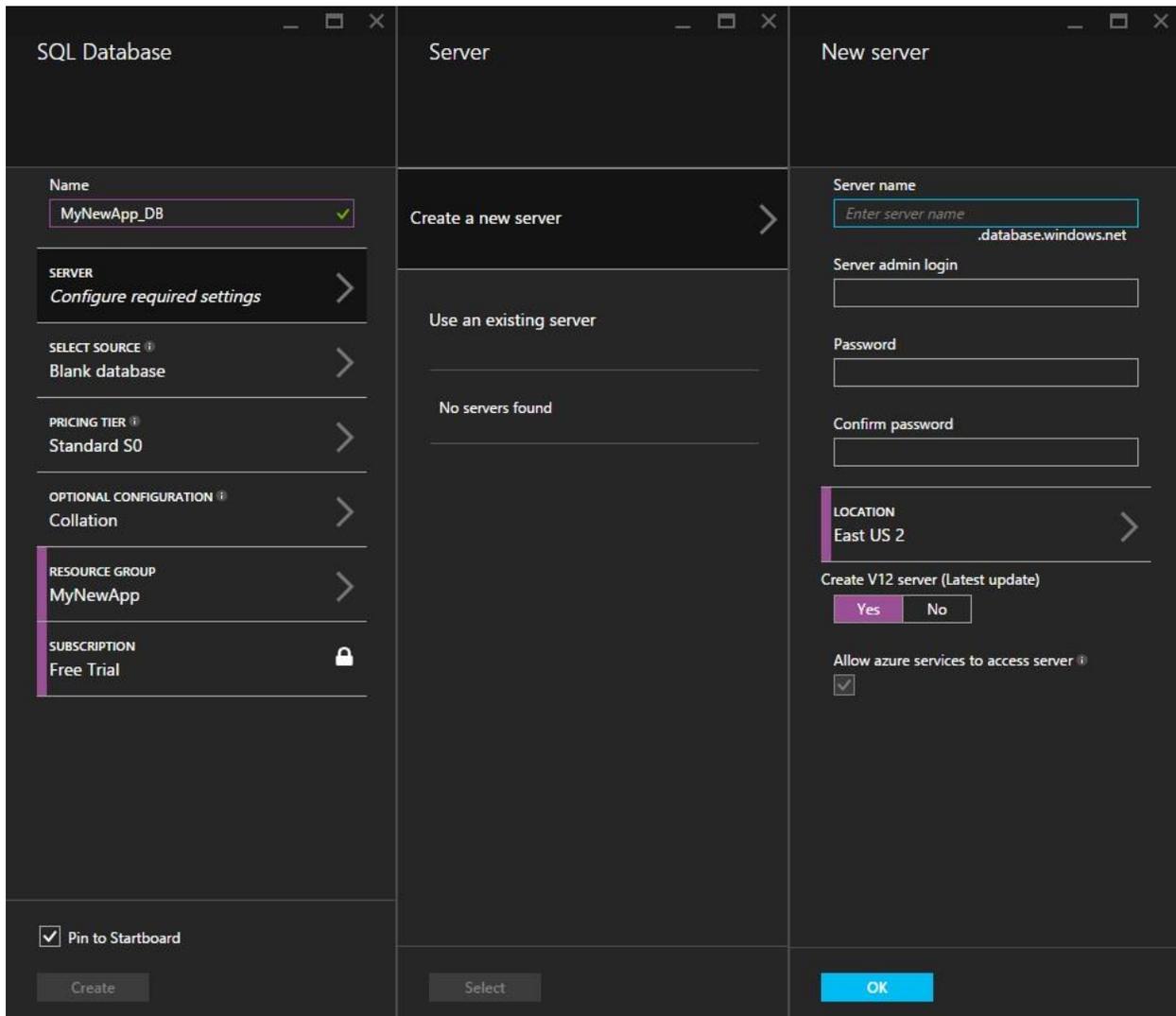
Next up is the pricing tier (note that at the top right of the pricing tier selections may be a Recommended | View All selection... click View All if not already selected). This isn't super critical right now as it can easily be changed later. Pricing and tier needs are hard to guess without watching them for a bit first... user counts, data storage, whether you'll use replication, stuff like that all affects the bottom line, so start small and scale up if you need to. S0 is a decent place to start for small-ish



applications with plenty of leg room and some nice features such as geo-replication that Basic doesn't have. Web and Business are retired and as of September 2015 (I believe) will no longer be available. Sometimes, such as going through heavy data migration periods, we'll ramp up the pricing tier a bit to get some more throughput, then knock it back down when we're done, so you can really pick any plan you want to start with as it only takes about 5 minutes for pricing tier changes to go in affect.

Ok next up is selecting a source for the DB. I typically use Blank (create an empty one), but you can choose the AdventureWorks sample if you want to, or you can load from a BACPAK data-tier application (but only one previously saved from Azure to an Azure Storage account... more on that later).

Toss in a database name (MyNewApp\_DB, or something) and then click the Server options to configure the server it'll sit on...



(see how that subscription is now locked on the SQL Database blade? Tricky bastards...)

Ok, a server name. Something that makes sense, maybe specific to the company or the project or whatever makes sense for whatever SQL Databases you might host on it (you can have more than one server later, so just use your better judgement as you would with an on-premise SQL Server instance).

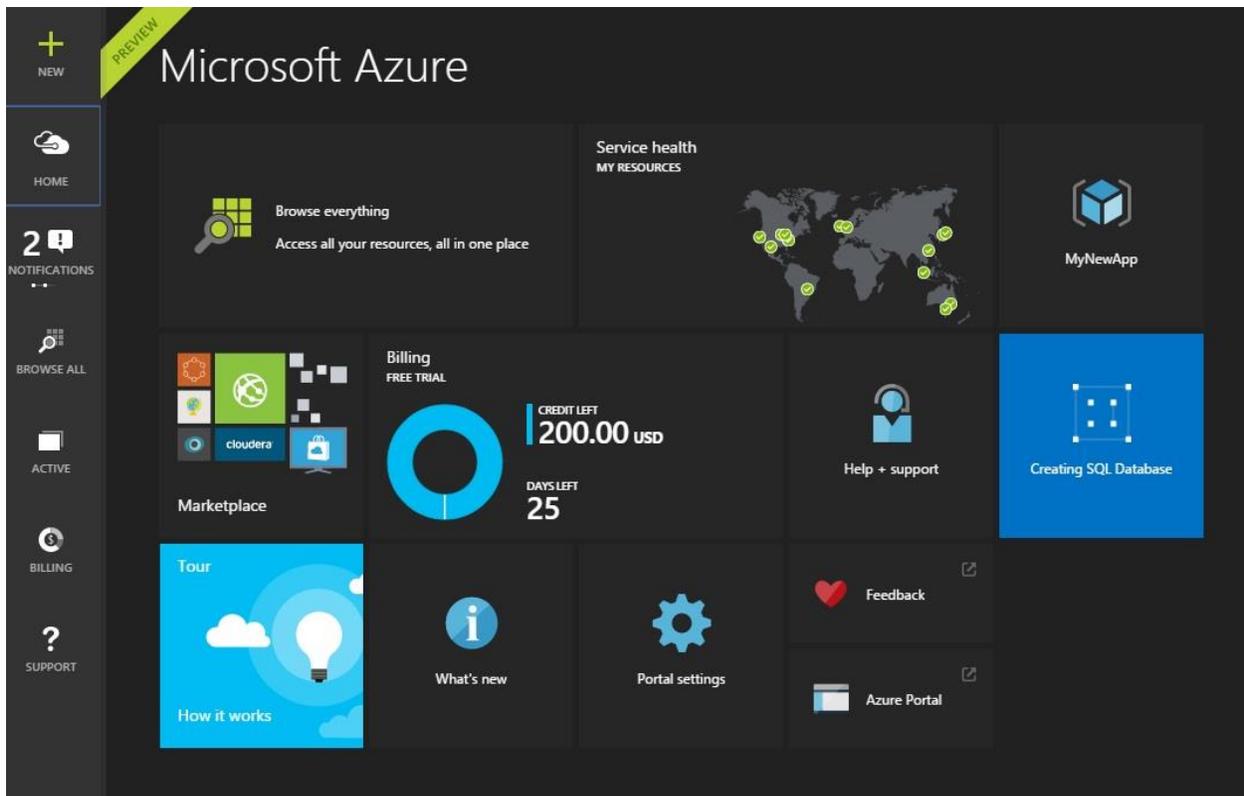
For the server name, it used to be that Azure would give you a string of random gibberish and that was your server name and you were stuck with it. Now we get to name our own, which makes things a tad more nice.

Server Admin Login: this is the master account for the database, which will log into the master database and be used to set up other users. I much recommend that this be a “set and forget” case and that you create a separate “practical use” dbo\_owner login for doing actual work.

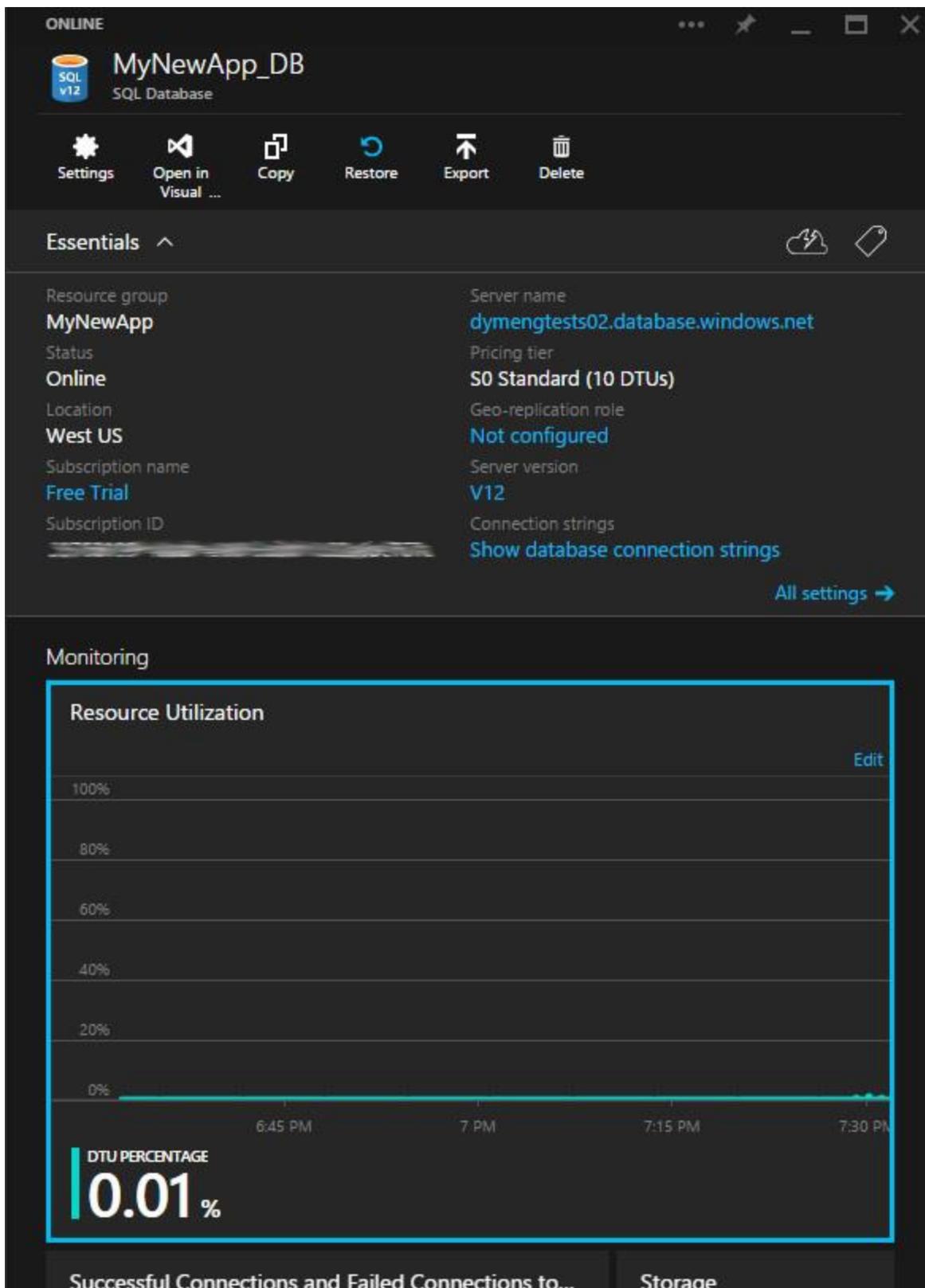
Choose a location (typically that’ll match your Resource Group location), choose your server type (V12 or not... I recommend V12 as the newer one, it has a number of features/services available that the older V2 version didn’t (better auditing, dynamic data masking, etc)).

Click OK to create the server, then create Create for the SQL Database now that all settings are complete (I like to pin this to the startboard as well, being that I tend to look at the db fairly often).

This takes you back to the portal home/startboard where you can see that the database is being created. Give it a few minutes to provision itself. In the image below you’ll see we have easy access to the MyNewApp resource group on the right, and will shortly have easy access to the database as well (these can all be found under the Browse All list on the left jumpbar as well, but that takes a bit of getting used to).



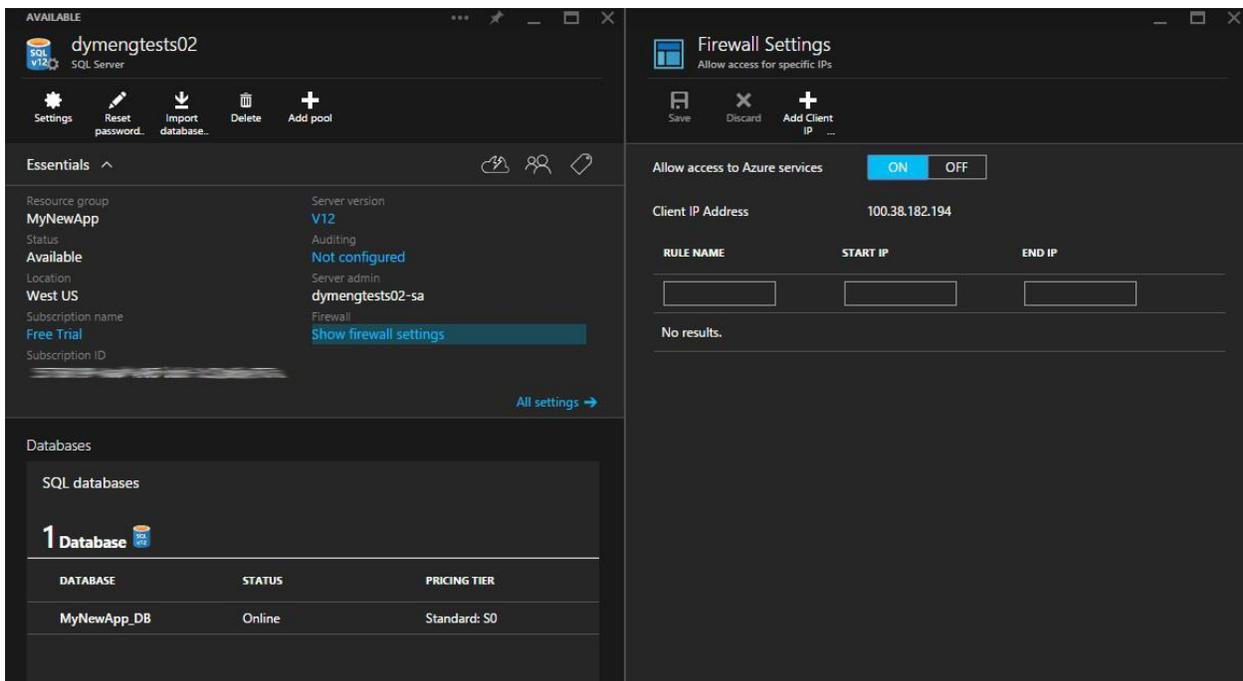
There we go, we have a database! Now let's get to some details... from the startboard either click the new database icon or the resource group. If you open the resource group you'll see the database listed, just click on that db (and the portal might open this blade automatically anyone once it's created). Anyway, find your way to the database details:



You have a detail section at the top and a scrollable section at the bottom. Check the detail section for Show database connection strings, you'll want those. I like to take this Resource Utilization chart and

pin it to the homepage during initial setup, I usually keep an eye on it to get a feel for where I am. Click the Edit button to add a few more data series to it if you like, then right click the chart and select Pin to Startboard (you can later customize the startboard by right-clicking the startboard and resizing/repositioning items). If you click on the actual chart itself, it opens in a detail with average, mins and maxes listed for each series. Handy.

Ok, one last thing we need to do... whitelist our IP address for the SQL Server instance. Click Browse All on the left jump bar and scroll down to SQL Servers, select the server we just made from the list, then click Show Firewall Settings, then add your IP address, a meaningful name and click the Save button/icon at the top of the Firewall Settings blade. (enter the same IP in both Start IP and End IP fields to whitelist a single IP address, provide a range (useful for internal networks) or if you want you can whitelist the whole internet if you want, but I wouldn't really recommend that (this does bring up the question of how to deal with desktop apps connecting via dynamic locations, which I'll address in a later article).



June 22<sup>nd</sup>, 2015

Jack D. Leach

**Dymeng**  
SERVICES   
<http://www.dymeng.com/azure>

