In order to follow along and perform some of these labs, this section will explain how to setup your database environment to allow you to proceed with future labs.

We will be using the PartsUnlimited database, which is created as a part of deploying the [PartsUnlimited demo application](https://github.com/Microsoft/PartsUnlimited/tree/master/) to an Azure web application and Azure SQL Database. You can read about the process here, but in order to keep this simple and scoped to the database only, I have extracted the database as a .bacpac file you can use.

## Requirements

In order to complete the labs yourself, you will need the following:

* Visualstudio.com account – You can sign up for a free one at <https://www.visualstudio.com/>
* Windows host computer with SQL Server 2016 installed and administrative privileges
* Visual Studio Enterprise 2017 installed – The evaluation edition works, and you can download this from: <https://www.visualstudio.com/vs/visual-studio-2017-rc/>
* SQL Server Management Studio 16.x – A free download from <https://docs.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms>
* The tSQLt testing framework, a free download from: <http://tsqlt.org/>

In addition, if you wish to deploy your database changes to an Azure SQL Database, you will need to have an Azure subscription to create the databases.

## Creating the Development Database

We will create a development database from the .bappac that will be used to perform all our development. We will also create two additional databases that we will deploy our database changes to. These will be blank databases, and can be created locally or in the Azure cloud. The labs will use the Azure cloud for the demonstrations.

Importing a .bappac is fairly simple, and there are instructions for doing so in the following post:

Once the database is imported, you have a development database. For the labs, I will use the PartsUnlimited\_Dev name to delineate that this is the development database.

## Create Downstream Databases

We could include programmatic instructions for creating the test and production databases, but this typically isn’t something done in a database development process. Since it’s a one time operation, it’s easier to create the database and have it ready for deployment. This also allows us to be sure security is different and more stringent in downstream environments.

In the labs, we will show these databases as Azure SQL Databases, with the names of PartsUnlimited\_QA and PartsUnlimited, for testing and production, respectively. If you choose to create these on your local SQL Server installation, you will need to adjust the paths and security accordingly.

## Setup VisualStudio.com

The last part of the lab setup is to ensure you have a VisualStudio.com account and a local agent. We use a local agent as the setup is easier than a hosted agent for deployment.

Signing up for VisualStudio.com is easy. Once you have a verified account, you can go to the account overview (<https://way0utwest.visualstudio.com/_admin>) and then select the “Agent Pools” item. Under this set of options, you should see a default and a hosted pool. We are interested in the “Download Agent” button. This downloads a .zip file of all the items you need to run a local agent. The Configureagent.cmd file will let you build an agent profile for your system. While you can run the agent as a service, I run it interactively so that I can control when it’s running and see all of the agent output.

Once configured, runagent.cmd is used to start the agent. I have a walkthrough of this on my blog at: <https://voiceofthedba.com/2016/07/18/building-locally-from-vsts/>