Azure cloud accounts

Microsoft has set up "Lab" accounts using a sponsored subscription to support investigation and development for this program.

- Connect & Accept lab 'assignments' from inbox.

Level-set.. all can login and view the prescribed budget

"Cloud-ify" the Database

During our last lab exercise, I created a database on the Azure platform, we replaced a single connection string in the local code base and we all contributed to building up a common, shared database which instantly affected the output of the activity for everyone with no noticeable performance impact.

- Create a new database instance

Try:

From the portal, Add a new database, Select the correct subscription, Create a new database server.

### *this location is not available for subscription*

Change location....

### *all locations selected return this error!*

Solution

Focusing on the technology, this *encapsulated* PaaS database solution is a pre-configured virtual system image which is essentially a SQL database service deployed on an operating system, which supports network access to the service over TCP/IP and safeguarded by a network firewall and user authentication.

- encapsulated pre-configured solution
- deployed virtual system
- supports TCP/IP access
- supports SQL Server *service*
- fire-walled
- authenticated login

  - Create VM, install and configure SQL Server
    - Configure and deploy the Virtual Machine
    - Configure the network firewall
    - Install SQL Server (on VM)
    - Enable TCP access to SQL Server
    - Enable SQL Authentication on SQL Server
    - Create admin login on SQL Server
    - Configure the operating system firewall
    - *Remotely connect, create database, tables, add data and update connection strings!*

Try:

[*https://github.com/uid100/Deploy-SQLServer-on-Azure-VM (Links to an external site.)*](https://github.com/uid100/Deploy-SQLServer-on-Azure-VM)

Security Considerations

When the appsettings.json file was updated, it included the connection string for a cloud-deployed, shared database and the connection string included authentication details which should not be stored to a GitHub repository.

There are a few good solutions for this vulnerability. This should be moved out of the code! But for now, as a short-term fix, keep the appsettings file out of GitHub.

Right-click the solution (not the project) in the Solution Explorer

Add > Existing Item and select .gitignore

Open the file and add save these lines to the file:

```
# ignore appsettings configuration files
**/appsettings.json

**/appsettings.development.json

**/appsettings.staging.json

**/appsettings.production.json
```

GitIgnore Patterns:

https://www.atlassian.com/git/tutorials/saving-changes/gitignore

When project (and solution) changes are saved to GitHub, this file won't be included. Previous versions of the file which are already stored in the repo won't be removed, so check to see that those files don't contain sensitive information. Consider removing them (their history is saved) and changing compromised passwords.

https://www.humankode.com/asp-net-core/asp-net-core-configuration-best-practices-for-keeping-secrets-out-of-source-control

---

Publish App Service

The SportsStore app service employs a layered approach to a web application solution. The textbook walks through a hands-on demonstration and evolution of these technology layer by layer and feature by feature.

Also, as a class, just like creating a *'hello world'* console app without writing any code almost in our sleep, we have demonstrated that using the pre-packaged template solutions that are configured into Visual Studio, a working (MVC pattern) web application is only a few clicks away.

We've discussed and investigated and resolved some of the many things which might go wrong with getting these applications running, how to debug and troubleshoot them, but even though we've connected them to a remote database by changing the connection string, the application is still relying on standing up a hosting service running on the local machine. A web application needs to be accessible from another platform, running on another computer, and maybe even from another network.

Just like the '*hello world*' solution and the packaged cloud database solution, *VS* provides a simple solution to push a working application to the cloud in as little as 4-clicks!

Try:

- https://github.com/uid100/Publish-AppService-to-Azure (Links to an external site.)