# Understanding the Difference Between Microsoft Azure and Amazon AWS

This course was built in 2016, which means it may be a little bit out-of-date.

## The Cloud

This course deals with the Cloud and major corporations – AWS, Microsoft, Google, . . . that provide a bundle of services. In the years shortly after 2000 clients (typically businesses) housed their web sites with companies who rented memory space in internet warehouses - storage for internet software and the business data that was pertinent. What makes the Cloud offerings different from these warehouses?

* Elasticity – dynamic adjustment to service demand – for short durations – and then retract when the demand returns to normal.
* Scalability – when the client’s business grows, it’s less painful to adjust
* Pooling – opportunity to tap into shared resources
* Service Provisioning – various employees of the client have different abilities to obtain access to services; the instructor referred to Role Based Access Control (RBAC)

Vocabulary

* **SaaS** - Software-as-a-service. **Word** and **Excel**, for example, housed in the cloud (no longer purchased nor residing on the client’s computer).
* **PaaS** - Platform-as-a-service. Tools in the cloud for software development.
* **IaaS** - Infrastructure-as-a-service. Tools in the cloud for system administration.

It is not too surprising that both Azure and AWS offer a free taste of their services. The amounts quoted (as of 2016) appear to be generous.

## AWS Free Tier

* Start at **aws.amazon.com**.
* The link for AWS Free Tier is under the **Pricing** tab.

The wizard is self-explanatory. Select **Basic** support to avoid being billed for training.

## Azure Free Trial Period

* Start at **azure.microsoft.com**.
* The link for free services is on the main page – the button labeled **Start Free**.

The trial period is one month, and the amount of free service is $200.

In 2016 the web site says “No commitment – trial does not automatically upgrade to a paid subscription”; as of May 2020, this may no longer be the case.

* In a conversation with a salesman, I was told “when the $200 or the 1-month trial period expires, I would start getting charges on the credit card.”
* But at <https://azure.microsoft.com/en-us/free/> (near the bottom of the page under “What else do I need to know?” the page contains **What happens once I use my $200 free credit or I’m at the end of 30 days?** And the answer that was displayed contains

We’ll notify you so you can decide if you want to upgrade to pay-as-you-go pricing and remove the spending limit. If you do, you’ll have access to free products. If you don’t, your account and products will be disabled, and you'll need to upgrade to resume usage.

On 5/18/2020 I spoke with Azure sales a 2nd time and asked them to clarify. The answer - from an Azure pre-sales representative - was that the account can be cancelled, and I will not be billed.

## Explore AWS

The first part of this exercise is to upgrade Visual Studio, so that it contains the extensions for AWS.

* Go to the web page <https://aws.amazon.com/visualstudio/>
* In the upper right-hand corner there are various download buttons. I clicked **AWS Toolkit for Visual Studio 2017 and 2019**. AWS responded by depositing **AWSToolkitPackage.vsix** into my computer’s Downloads folder.
* Make certain that Visual Studio is not running. To be safe exit all programs running on the computer.
* Double-click **AWSToolkitPackage.vsix** in the Downloads folder. This installs AWS-interface software into Visual Studio. The execution time is fairly long (approximately 20 minutes).
* Start Visual Studio. Open a solution that has been tested in **wwwroot**.
* Click the **Create a new project** in the right-hand pane.
* Type “AWS” into the search textbox at top center of the page.
* Visual Studio should display several templates pertaining to AWS. This confirms that the AWS Toolkit for Visual Studio has been successfully installed.

The next part of this exercise is to link Visual Studio with AWS.

* Start at the AWS Web portal.
* Select the geographic region – in upper-right corner near the account identifier. AWS displays a drop-down list. Choose a close region.
* Click **Services** – in upper-left corner. AWS displays a page full of services. Select **Elastic Beanstalk** (in the **Compute** category). AWS displays a web page titled **AWS Elastic Beanstalk**.
* Click the **Get Started** command button (titled**Create Application**). AWS responds by displaying a new web page titled **Create a web app**. Type an appropriate name into the **Application name** text box.
* Tags and associated values are optional. I wrote some indication that this is the 1st test application.

The next part of this exercise is to create an application source bundle for AWS.

* Start a Visual Studio web-application project that has been tested in the wwwroot folder.
* Right-click the project node in the solution explorer.
* Click the **Package/Publish Web** tab in the left column.
* In the **Items to deploy** section, select **All Files in the Project Folder** from the drop-down list.
* We want to deploy to AWS’s **Elastic Beanstalk**. Right-click the project node in the solution explorer. Then click **Publish to AWS Elastic Beanstalk** from the drop-down list. AWS displays a pop-up dialog titled **Publish to AWS Elastic Beanstalk**.

At this point I am stuck. There are other steps that are evidently needed before I can publish a self-generated web application to AWS. I will probably need more Pluralsight training. The instructor of this course recommended two follow-on courses.

**Modernizing Your Website with Azure Platform as a Service**

And

**AWS Developer: Getting Started**

## Remainder of this Pluralsight Course

The remaining sections of this Pluralsight course are

* The IaaS Scenario
* The Hybrid Cloud/Identity Scenario
* The SaaS Integration Scenario
* The Cloud Storage Scenario
* The Database Scenario
* The Security Scenario
* The Availability/Disaster Recovery Scenario
* The Decision Point

I started listening to the 1st of the above (The IaaS Scemario), but it became quickly apparent that (1) it was beyond my level of experience, and (2) it was probably much more pertinent to businesses larger than Current Pixel. Therefore, I skipped all of those sections except **The Database Scenario** (DBaaS).

#### With regard to DBaaS (Data as a Service)

There is an old-versus-new competition. (Refer to <https://www.petri.com/azure-web-applications-iaas-versus-paas>.)

The old: IaaS (Infrastructure as a Service) (simply stated “virtual computers”) used to hold the executable web-application code and the corresponding database(s).

The new: PaaS (Platform as a Service) (contains web services) in conjunction with Data Storage (which comes with a few other useful services).

It appears that – for most businesses – using either AWS or Azure - **new** wins over **old** (cost and benefit).

SideNote While listening to the discussion, it finally dawned on me what “database deployment” means. Having been recently been introduced to web-application projects, where the database is a SQL Server LocalDB (a .mdb file) that is part of the project, I got the false impression that one might simply upload the entire Visual Studio project to deploy both the executable code and the database. WRONG. At this point I have only a fuzzy picture of what database deployment entails. I presume that a separate set of instructions (or possibly a console batch file) would be needed to upload the schema (or changes to the schema). We’ll see.

If Current Pixel settles on SQL Server, Azure might appear to have a slight edge over AWS, because SQL Server is a Microsoft product. AWS does supply database management services for a variety of databases including SQL Server, but for most of these, AWS has a bring-your-own-license policy. However, SQL Server Express is an exception, and AWS waives the license requirement for this type of database.

From the Developer’s point of view: if the training has been primarily .NET and SQL Server, then Azure is the natural choice. On the other hand if the training has been with open-source products (LAMP: Linux, Apache, MySQL, PHP), then you might be better off with AWS. Keep, in mind, however there are other points of view and reasons that might tilt in favor of AWS (cost, scalability, security, etc.)