

Choosing your cloud platform

Date: 2020-09-14 10:20:00

I cover all of the aspects, at a high level, that you need to consider when developing, finding a cloud provider, getting a web domain and securing your website, ordered by the main cloud providers in terms of technical complexity required to get started.

First some website development basics

Creating web content

Web content tends to be a mixture of:-

[Hypertext Markup Language \(HTML\)](#) - Current version of this is known as HTML5. This is where you write the bulk of your content.

[Cascading Style Sheets \(CSS\)](#) - This provides the styling e.g. you can define the colour & fontsize that you want for a heading separately from that which you want for the body. CSS files are usually stored separately from the HTML web page files and referenced by them. CSS is really where you'll need to employ somebody with an artistic flair, if you want your website to look cool. Like HTML, CSS comes in versions, with the latest one being CSS3

[Javascript](#) - This allows code to be called from the web page, typically as the result of an event or action e.g. if you don't fill in the correct content in a field, you call javascript code to inform the user that the field needs to be completed. Javascript code can be found easily by googling and there are plenty of examples as to how to incorporate javascript in to your webpage to do a variety of actions.

Static website

Most people start by creating a website which just has information on it. As the content remains the same as when you first publish the website, this is known as a static website. Even webpages that incorporate javascript code are still called static websites. To create nice looking websites, typical choices are:- 1. [Wordpress.com](#) 2. [Wix.com](#)

Software as a Service (SaaS)

Rather than building everything yourself from scratch, it's much easier to integrate with SaaS providers e.g. you might want to integrate with a payments gateway e.g. Paypal to provide a checkout service for your website, or Disqus to add comments to your blog

Dynamic websites

Dynamic websites differ in that they rely on server-side code (Common ones are PHP, Node.JS & Python) to generate webpages on the fly. This means that dependent on what the user chooses or their settings, different webpages can be returned

Web Hosting

In order for anybody to be able to access your website, you'll need to have it hosted by a cloud provider. This means that your website html5, css, javascript (and server-side code, if relevant) needs to be uploaded to a cloud web hosting provider.

Web Domains

When you first develop your site, you'll typically be given a technical looking url e.g. <https://deytalytics.github.io/deyblog> provided by the cloud provider In order to get a nice looking url that's more relevant to your business e.g. <https://www.deytalytics.com>, you will need to separately purchase a web domain. Web domains are typically rented on a yearly basis and can cost very little. I think I pay £10/year for the deytalytics.com domain.

Website Security

If you're just creating a website for your own needs, then website security isn't top of your mind, but nowadays, if you do anything much more sophisticated, you can find yourself in a legal minefield if you're not careful. For example, if you wish to store a database of people who've shown an interest in your product, you need to ensure that no hacker can steal this personal info. For these reasons, you'll eventually need to set up website security which will ensure that only the people that you want to can do things like change the website content or access a backend database. In addition to this as your needs progress, you'll want to add in logging functionality to audit anybody whose trying to access your website and prevent denial of service (DDOS) or code injection attacks

Website provider choices

Common Options for Individuals & Small Businesses

Although an individual or small business may choose to use any cloud provider due to having smaller budgets than larger businesses, the options below are the typical choices made.

[Wordpress.com](#)

Wordpress is pretty much the go-to site for anybody who wants to produce a static website with a rich set of functionality provided e.g. styling/themes and plugins which allow you to integrate with a vast range of other tools. Wordpress.com is the version of Wordpress where they'll host your site for you. Wordpress.com allows you to develop your website on their website.

Pros:

Simple to get started, deploy a site and get a web domain.

There are a vast array of plugins on offer that allow integration with most well known SaaS providers

Cons:

Can be difficult to differentiate your website from 100s of others, as most people don't do much to change the theme.

Wordpress.com control everything about your website. If their business fails, potentially you could lose your website.

As your needs become more sophisticated e.g. you might want improved security or more control over who can access the site, Wordpress becomes more limiting, and you'll need to migrate to another provider.

[Wordpress.org](https://wordpress.org)

Wordpress.org allows you to download the code that is used to build your website and develop your website locally. You'll then have a choice as to which web domain & cloud provider you wish to use.

Pros:

Gives you all of the functionality of wordpress.com but you own your own website material and can choose your own web host & domain

You aren't sharing your website with 100s of 1000s of other people's websites so the performance can be better

As you can pick & choose your web host & web domain provider, you may be able to save money

Cons:

Requires more technical expertise than Wordpress.com

Although you have more control over how you can adapt your website, you're still dependent on Wordpress for any plugins you might use.

[GoDaddy](https://GoDaddy.com)

GoDaddy started off by being best known for selling web domains. They do provide a web hosting platform as well, however.

Pros:

1. It's less expensive than going for a more fully functioned cloud provider

Cons:

It's very limited on being able to integrate with other software providers that are not hosted by GoDaddy e.g. if you wanted Tableau to produce reports against your GoDaddy website, GoDaddy doesn't support Tableau and you need to open up ports, risking your security, in order to allow the integration.

If you're producing a dynamic website, you're pretty much limited to PHP which is considered an outdated web language nowadays

[Github pages](https://github.com)

Github is best known as the place to store your code/website material. Less well known is that they provide the ability to host a static website for free.

Pros:

It's free

As you should be storing your website code/material somewhere, GitHub is 1 of the best places, so it's not a huge learning curve to setup GitHub pages

Cons:

You're limited to a static website.

There's no assistance in creating your own website, so you'll have to be fairly technical to use this option.

[Heroku](https://Heroku.com)

Heroku is another developer friendly method of deploying code. The Pros & Cons are pretty much identical to Github pages, but an advantage of Heroku is that you can also deploy a dynamic website that uses PHP, Node.JS or Python. Heroku doesn't provide a https url on the free tier, but you can stick cloudflare.com

Big Boy Cloud Providers

For larger businesses, these are the 3 main cloud providers as of 2020 (Note: There are others like IBM, Oracle & Redhat Openshift but they're not popular) The main cloud providers really provide the following functionality that makes them different:-

Separation of the 3 tiers of website development (Presentation, Business Logic, Database)

Fine grained control of cloud security

Sophisticated logging tools

Ability to deploy dynamic websites using a variety of development languages

Containerisation of website applications - This is very useful to provide a reliable deployment for complex websites

Ability to be able to deploy an application from a local environment to the cloud using an API

Able to deploy your website application in different geographies

Able to control the scaling of the deployment environment to adjust for a cost vs performance budget model

Budget reports so you know how much you're being charged for each service.

[Google Cloud Platform \(GCP\)](https://GoogleCloudPlatform.com)

Google are obviously known for producing a range of funky web tools as well as their ubiquitous search engine. Google were relatively late to the party for enterprise scale cloud provision but produce a decent enough platform

Pros:

They do provide the functionality mentioned above

As well as the common functionality, Google can provide services such as a custom search of your website, integration with Google Maps and a Google Home Skills kit

Cons:

1. Google's products are less well known by end users than Microsoft's and are much less extensive than what AWS offers.

[Microsoft Azure](#)

Pros:

Microsoft have the leading database in the non-cloud world - SQL*Server. As it can be difficult to migrate from existing databases to new databases, Azure is often the choice made for this reason.

Azure integrates well with Office360 products

Cons:

1. Some Azure products can be pretty ropey e.g. Azure Data Factory V1 was little better than a job scheduler and didn't provide the data integration that enterprise level businesses require.

[Amazon Web Services \(AWS\)](#)

Amazon were the first to create an enterprise cloud platform and are still the cloud platform that very technical developers enjoy using the most.

Pros:

AWS is the most technically sophisticated of the 3 cloud providers providing services that cover pretty much anything you'd want to do technically.

As well as AWS, Amazon provide the Alexa Skills kit

Cons:

1. Technical sophistication comes with the need for more technically expert staff to be involved.