- Getting your Cloud App from the ideas stage to fully formed, written, and deployed is a long process, but there are many tools which will help you along the way.
- A cloud application developer's workbench includes:
    - Version Control,
    - Libraries
    - Frameworks
    -

Version Control

- When many developers are working on the same project, knowing what order changes were made, thereby creating a new version of the source code, becomes overridingly important.
- Version control systems keep track of what changes were made when and by whom and resolve any conflicts between changes.
- For developing your code version control can be useful even when you are the sole contributor on a project.
- Properly used, it can give you a way to revert to an older version of your code if something goes wrong and gives you some basic information about how the code developed over time.
- Version control functionality is generally tied to the storage system you are using, which is why a code repository is recommended, even for beginners.
- Git and GitHub are extremely popular for source code storage and management. Git stores files in repositories where you can track changes, split code into different branches for more focused development, and then merge them back into the main body of code.

Libraries

- Libraries are collections of code, like standard programs and subroutines, that you can use within your code.
- For example, you might want to include a navigational feature, like a carousel – a code library can supply you with the code for that so that you don't have to spend the time and energy creating one from scratch.

- Being able to reuse code in this way makes developing your app much quicker and easier. Multiple code libraries can be integrated into your existing project. As you discover a need for a specific function or feature, you can research an appropriate library.
- You determine when to call the required method as needed, and control returns to the program flow once the subroutine is finished.
- When you use a code library, you are in control. Code libraries are generally used to solve a specific problem or add a specific feature set.
- Either way, there are lots for you to choose from, so do your research.
- Here are some examples of code libraries:
    - jQuery is a JavaScript library that simplifies Dom manipulation.
    - Email-validator is a small library that checks an email address is valid
    - Apache Commons Proper is a repository of reusable Java components.

Frameworks

- Frameworks provide a standard way to build and deploy applications. You can think of a framework as being a skeleton that you can extend by adding your own code, providing a scaffold on which to build your apps.
- The framework you intend to use must be determined early in your development planning and used right from the beginning.
- New frameworks can't be incorporated into an existing project. Your chosen framework dictates the architecture of your program and controls the program flow.
- The framework determines which subroutines and methods will be called when. When working
- with a framework, there is a specific structure that you must follow. The framework is calling calls on your code, rather than you calling on the framework. Frameworks are less flexible than libraries, allowing you less control, but they do provide good standardization and can help you create efficient code.
- To use an analogy, if you are a carpenter building a house, the framework is the frame that you add to – bricks on the outside, plasterboard on the inside, and so on. The frame acts as a guide for how the house is constructed.
- Here are some examples of frameworks:

    - AngularJS is a JavaScript-based framework for dynamic web applications.
    - Vue.js is a JavaScript framework focused on the user interface.
    - Django is a framework that uses Python for web development.
- Frameworks define the workflow that you must follow, unlike libraries, which allow you to call functions as and when required.
- When using a framework, it can sometimes feel like you, as a developer, are not in full control of the development process. This sense of the framework and its predefined workflow controlling the development process is referred to as inversion of control.
- Frameworks that have a lot of control are known as opinionated – they have opinions on how their workflow should be used and remove a lot of the decisions you would otherwise have to make about how code is written, the location of files, and even file names.
- Frameworks often include their own libraries, which they call when needed. Inversion of control allows you to create standardized apps, and takes away a lot of the tedious configuration work, so you can focus on the code for your app.