When the ASP.NET CORE MVC Application is loaded several things happen behind the scene, you need to understand them all. Very necessary.

1. Default Web Host is created in which our application will be hosted. So the Hosting Environment is created with default settings. WebHost Class is responsible for creating the WebHost. This is the Hosting Env for our application. During Web Host creation, several things happened.
2. Configuration Sorces in asp.net mvc are appsettings, lauch settings, OS level env variables, User Secrets, these configuration sources hold info about the Web Host Configuraiton and Application Configuraiton, Previsouly we have Web.config that hold Web Host and Application Configuration.
3. During Default Web Host Creation, the Configuration Sources are read and Web Host is configured accordingly.
4. During Default Web Host creation, All Configuration Sources are read and all Config Variables are loaded in the IConfiguraitonService and provided in the StarupClass. This IConfigurationService is also available in the application as well. We can get this service by Constructor/Method/Property injection.
5. During Default Web Host creation, All Logging Sources are also configured and provided within the application as Single Variable as ILoggerServices I think. This logger variable is linked to all Logging Destination where we want to log info. We set Logging Configuration in some configuraitno sources.
6. Asp.net core mvc has built-in DI Container, that make such services availale in the application code.
7. Program.cs & Startup.cs are linked to each other. These both classes create the Web Hosting Env, Application RPP, and host the application code in the env.
8. Application RPP, is the more specific Application Hosting Env in which application will be hosted. Creating RPP contains 4 tasks.
   1. Adding the Default middleware to the RPP, in Configure()
   2. Configuring or customizing the Default middleware and then adding it to the RPP, in Configure()
   3. Adding the Default Middleware Services needed by the Middleware to the DI Container, in ConfigureServices()
   4. Configuring or customizing the Default Services needed by the Middleware and then adding them to the DI Container, in ConfigureServices()
9. Asp.net application container the Kestrer WebServer within the application. Method Package download the assemblies for the Kesteral web server. So asp.net mvc core application is complete Web Server + Web App Hosting RPP + Web App.
10. In Process hosting means hosting within IIS or IIS Express.
11. OutofProcess Hosting means host with in the Kestral Web Server.
12. Lauchsettings.json only matters in development env. Not relevant in Production.
13. Kestral web be Internat Facing Web Server or the Backend Server
14. Kestral is cross platform Web Server that can run on any OS.