There are multiple ways that deployment works from local to WWW and here we will try to explain it in a more generic way so that it helps you understand the process.

**How it works:**

Now let’s start step by step how the process works,

1. We start writing code on our system
2. We then build it and run it to test on the local server
3. Now we commit the changes and minify the code
4. After we commit the code we build the project with new code changes and put that build .WAR file in a machine we call server
5. Now just like we ran it on our local we our application in that machine(server)
6. This server will have an IP and a port on which the application runs
7. This IP is assigned a domain name like [WWW.ABC.COM](http://WWW.ABC.COM) using DNS
8. So when we hit that domain name we see that application running on that [WWW.ABC.COM](http://WWW.ABC.COM) instead of localhost:8080

And this is how the journey from localhost to WWW happens.

Now there are multiple ways to achieve this like using Docker, Cloud etc. but the defined process tries to explain the more generic scenario because in all cases it is the server that runs the application at some port and have some DNS with it to see the application on that.

Now, what about **when new updates are released?**

There is something called Pipelines that are created to automate the deployment of applications so that developers only has to take care of the code. Whenever a developer commits the code this pipeline runs automatically and reflect the changes on the application.

1. Make changes to the code
2. Commit changes
3. The pipeline builds the application and deploys it
4. The application refreshes and reflects the changes

Taking a specific examples of deployment using Docker and Kubernetes:

Commit code -> Git pipeline runs -> New Docker image pushed -> Kubernetes configuration and deploy