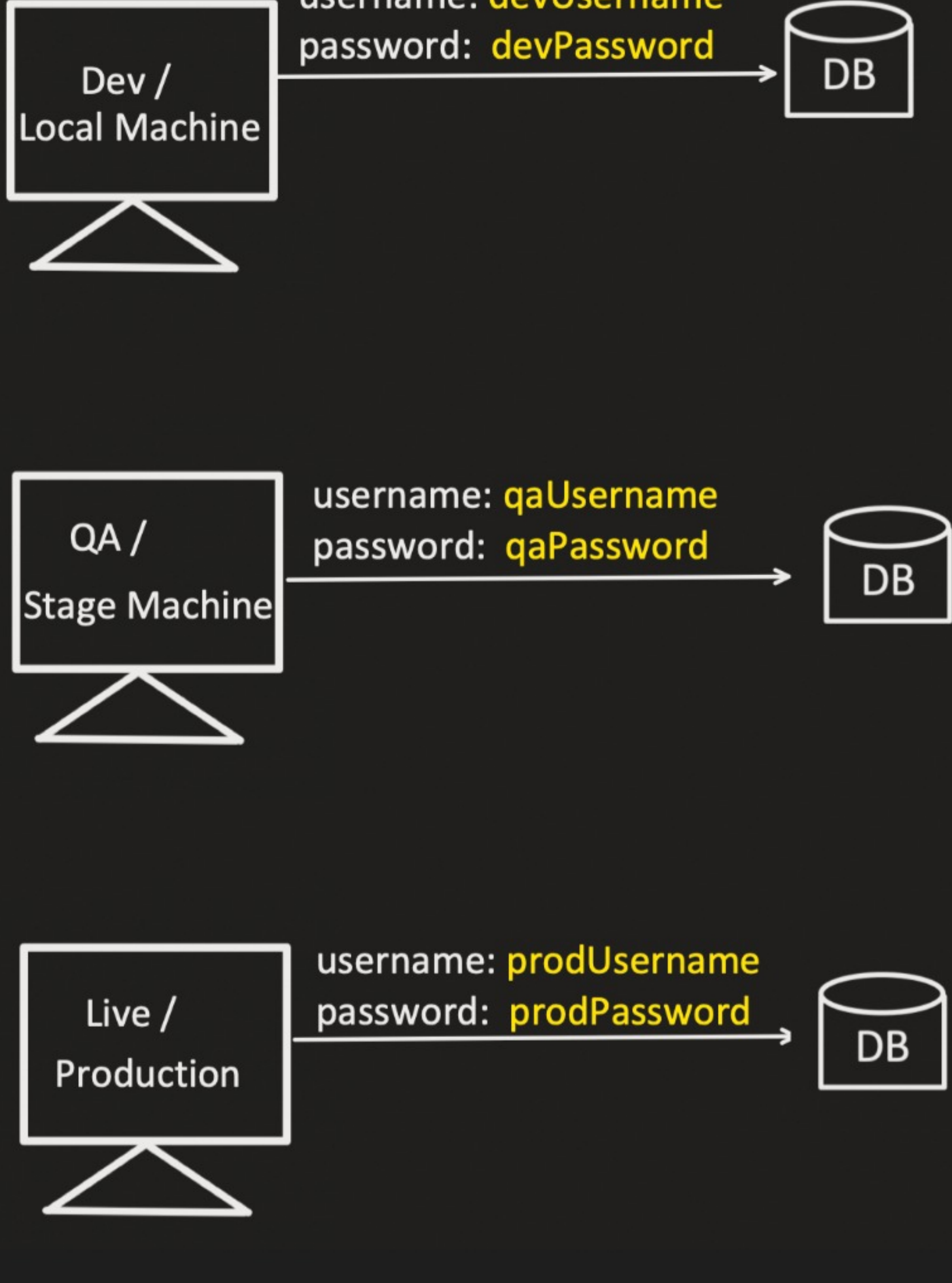


## Spring boot: @Profile

Let's deep dive into @Profile annotation to understand it better



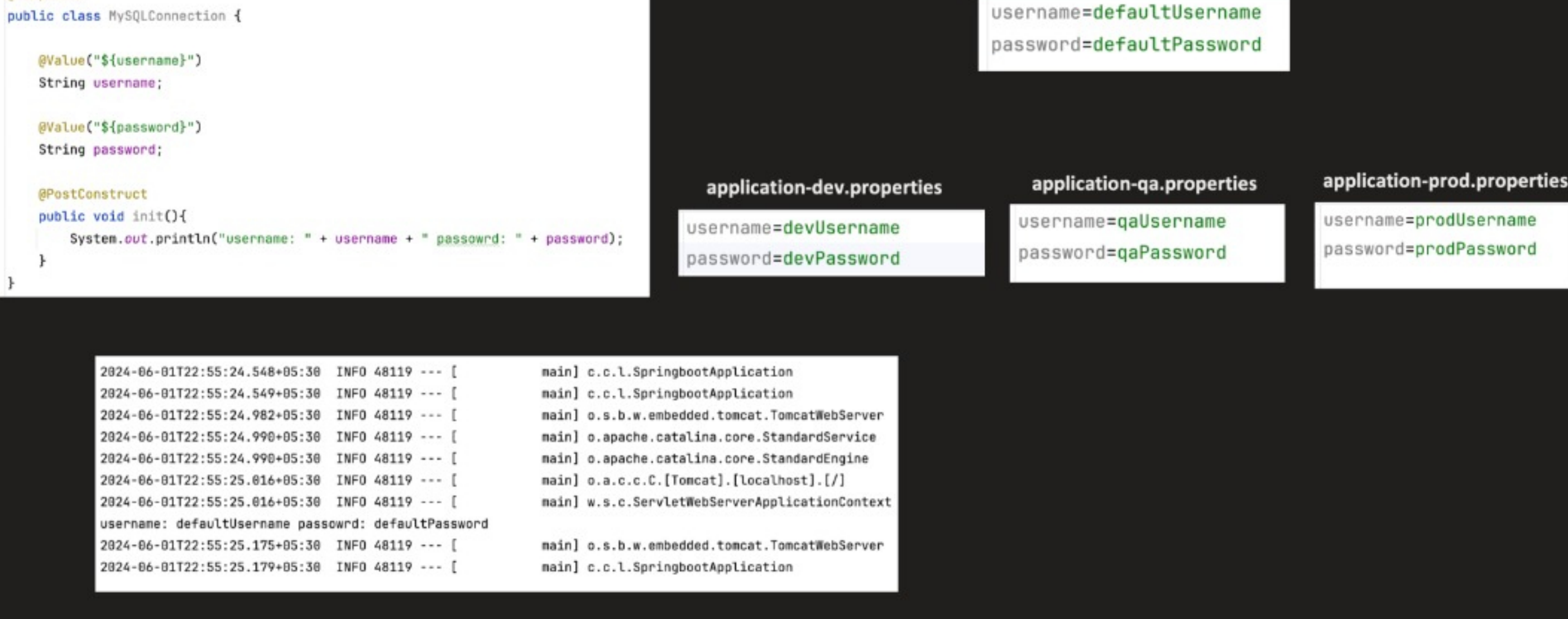
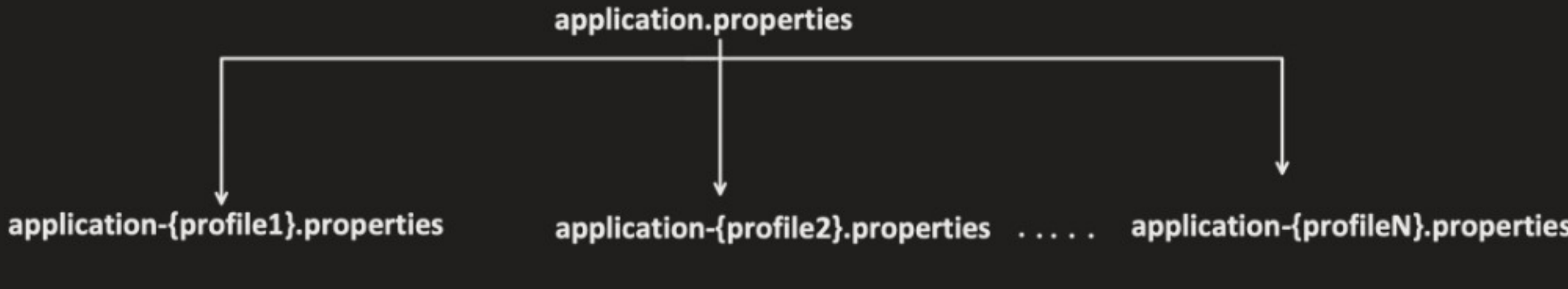
Report Abuse

This "username", "password" is just one example. There are so many other configuration, which are different for different environments, like:

- URL and Port number
- Connection timeout values
- Request timeout values
- Throttle values
- Retry values etc.

How to do it?  
We put the configurations in "application.properties" file.  
But how to handle, different environment configurations?

That's where Profiling comes into the picture

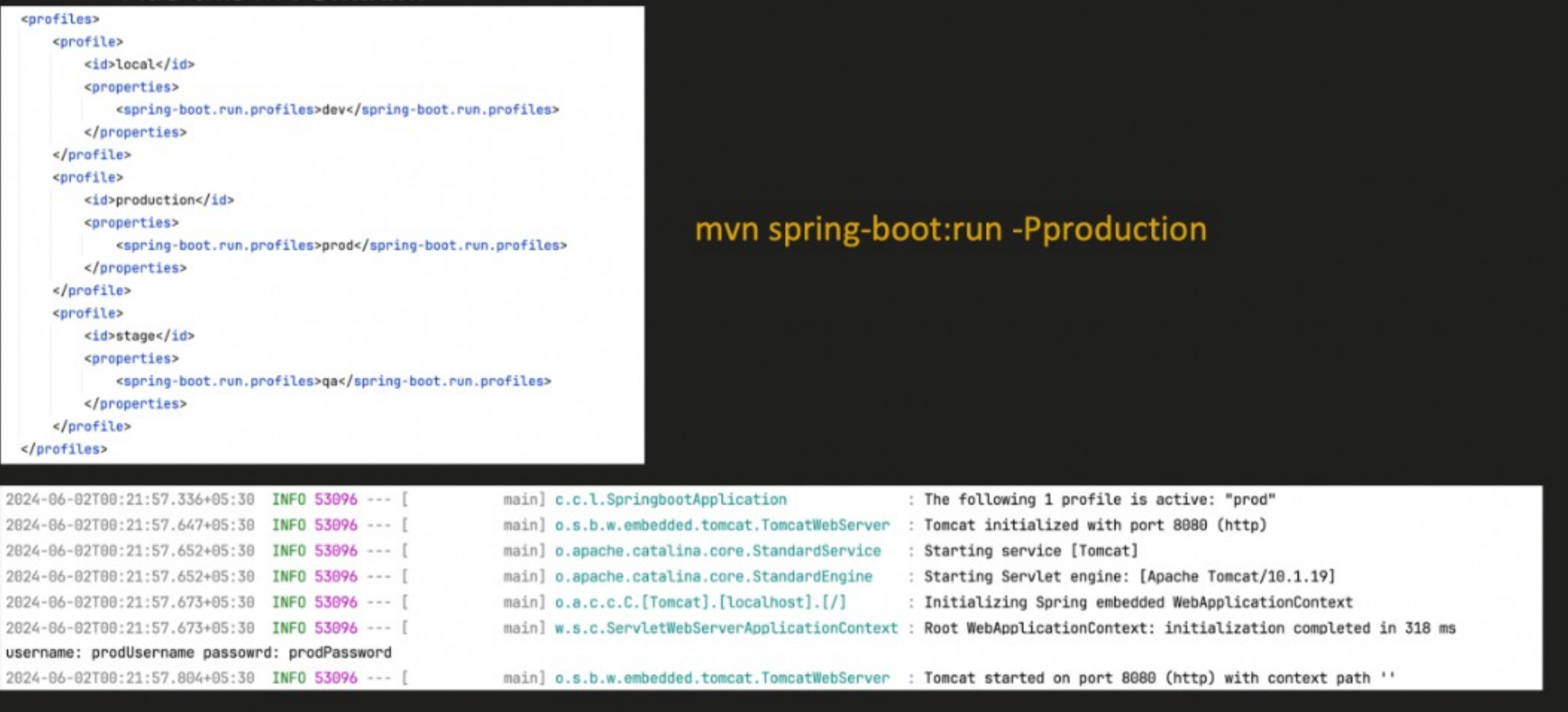


We can pass the value to this configuration "spring.profiles.active" during application startup itself.

mvn spring-boot:run -Dspring-boot.run.profiles=prod

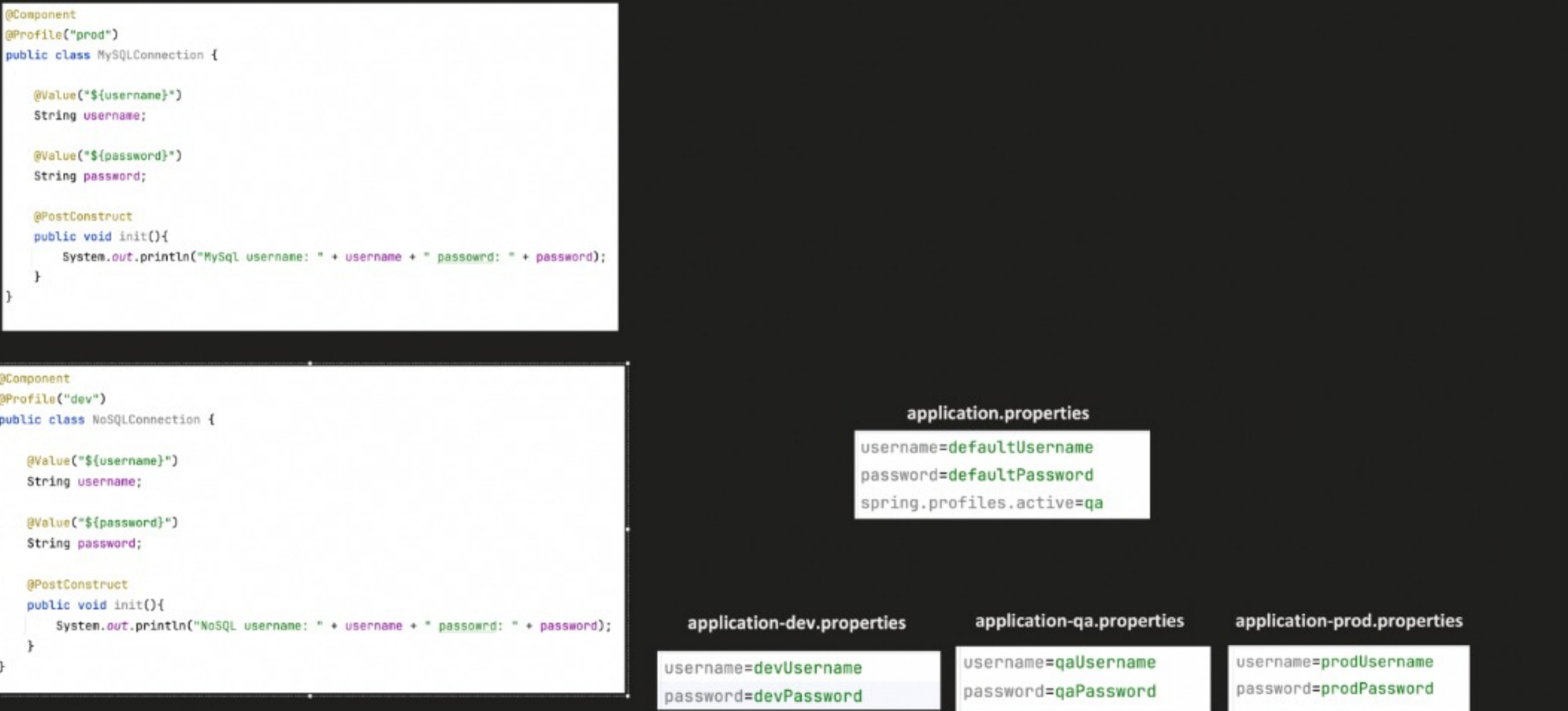
or

Add this in Pom.xml

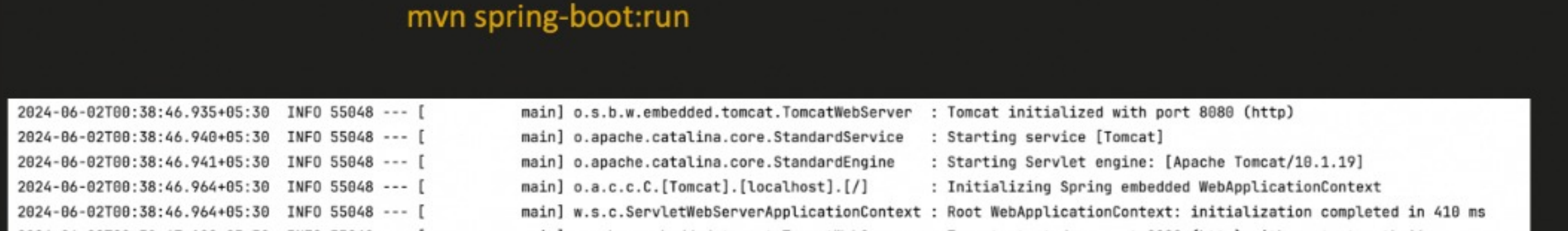


So, now we know, what's Profiling is. Lets see what is @Profile annotation.

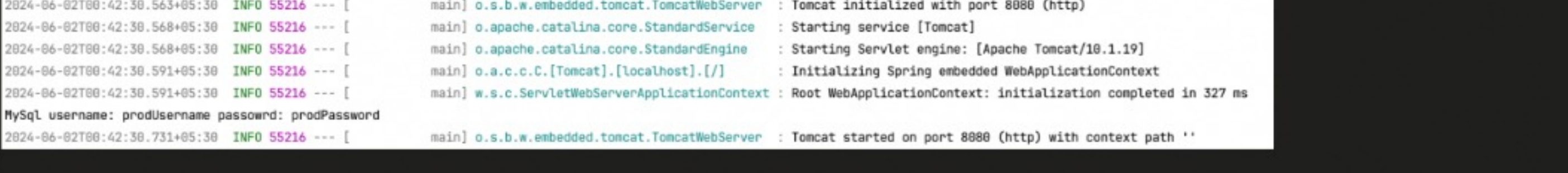
Using @Profile annotation, we can tell spring boot, to create bean only when particular profile is set.



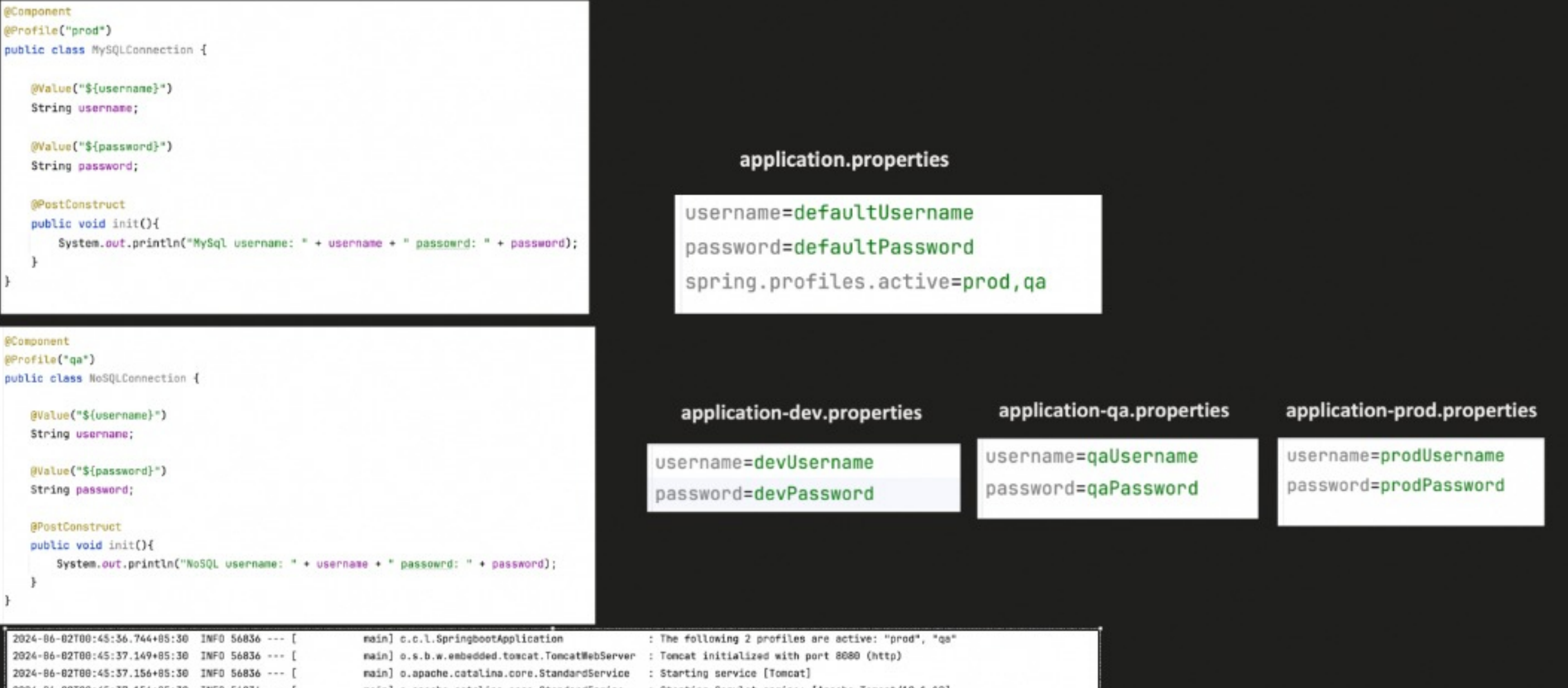
mvn spring-boot:run



mvn spring-boot:run -Dspring-boot.run.profiles=prod



We can also set multiple profiles at a time.



Now lets come back to the previous question, which I asked:

You have 2 Application and 1 common code base, how you will make sure that, BEAN is only created for 1 application, not for other?

Common Codebase

