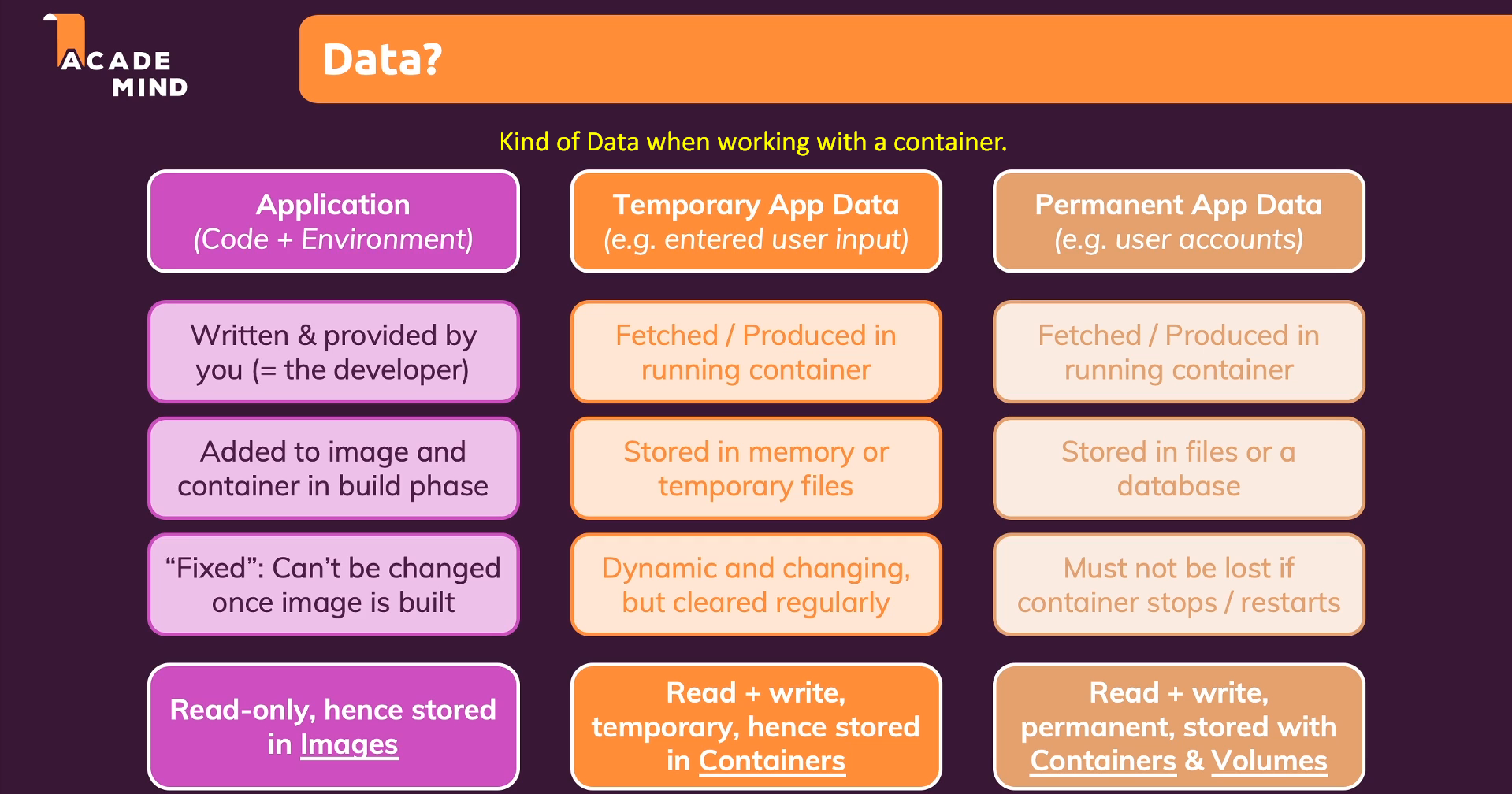
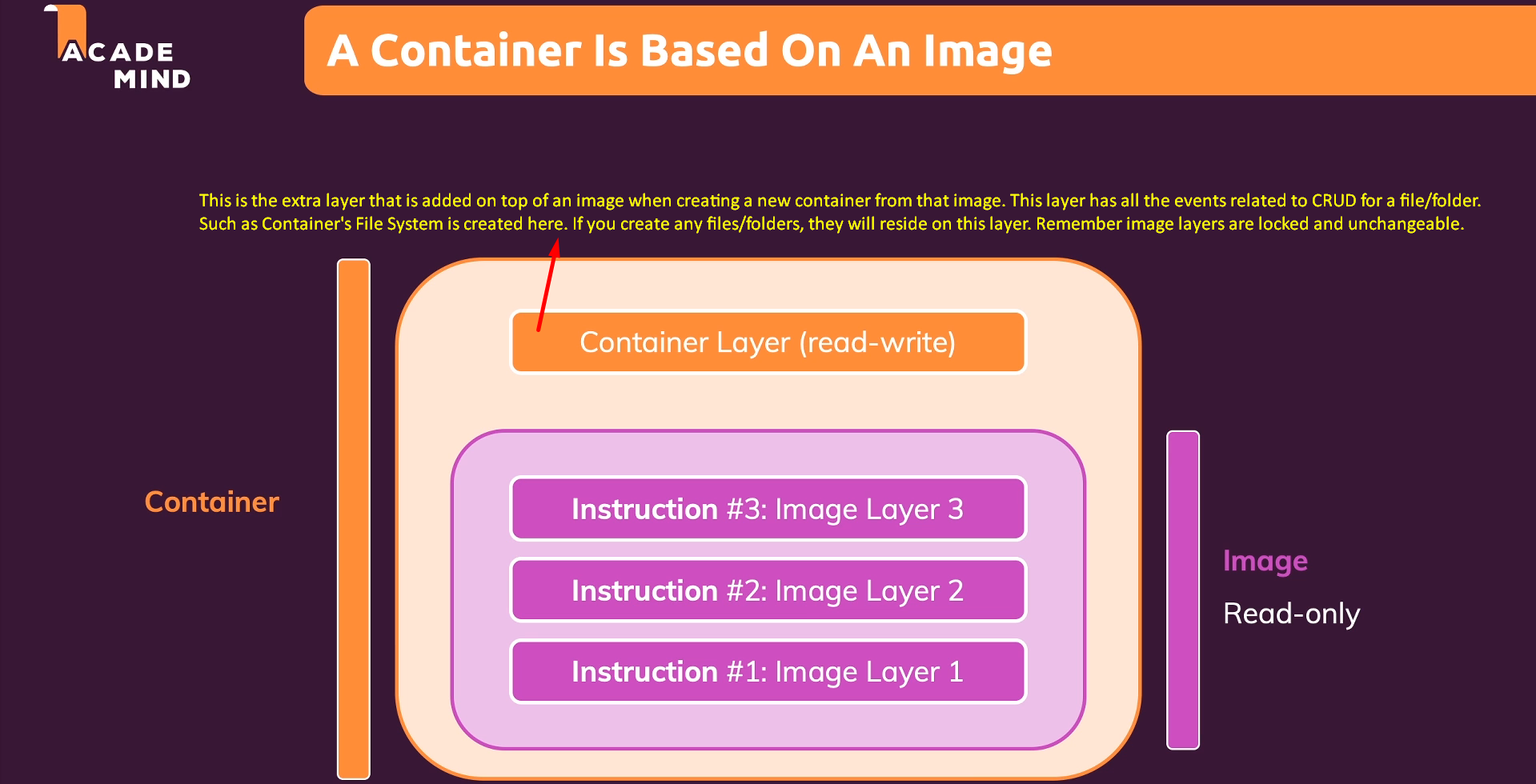
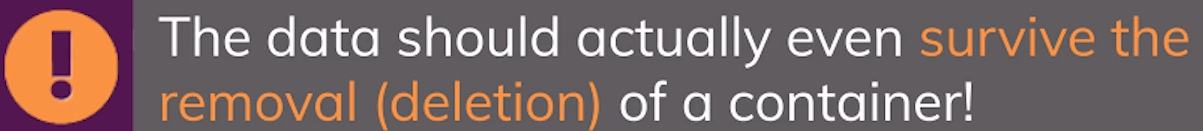
1. We need to understand what kinds of data we have in order to understand which kinds of problems we can face and what kinds of problems can be solved.
2. **Kinds of Data**:  
   
   1. **Application = Code + Environment**.
      1. So, we have an app which is source code and the environment in which it runs.
      2. This is what we talked about a lot in our last module.
      3. Some code parts are written by us and some are from 3rd parties like dependencies.
      4. Fixed: Now when we build an image, the complete source code (along with dependencies) and environment are copied into the image and the image is fixed which means whatever is inside the image can’t be changed.  
         When all the instructions in Dockerfile are executed, the image is locked.  
         Images are REAL-ONLY.
      5. **But this is not a problem as we also want our source code and environment must be read-only & that is why we copied into an image.  
         We don’t want our running app to edit source code & it is not right too.**
   2. **Temporary App Data**:
      1. Data Generated while the app is running such as the data entered by a user on a web site.
      2. Such data which is stored in local variables or **temporary files** in docker (We call it **temporary** because those files will be gone when the container is gone).  
         Such data and files are stored in extra layer of container. The layer which is added on top of an image when creating a new container. Jatin: Called **Command Layer**.  
         Extra layer means when we create a container from an image, an extra layer is put on top of the image which makes it a container.  
         Docker has **READ-WRITE access** on this layer.   
         Actually, this is where container’s file system is managed.  
         **NOTE**: Image has its own file system which is **unchangeable**.  
         **NOTE**: The final effective file system is driven after merging Docker’s file system and container’s file system.

  
Actually, there is file system on image itself which is fixed and whenever you do any changes in file system like creating/deleting files, they are done on container’s file system and eventually Docker combine both file systems to drive the final state of the file system.   
  
And this is not what we didn’t do up to now & this is what we’re going to do in this module.

* 1. **Permanent App Data**:
     1. When a user signs up and we want to persist that data in a file or DB.
     2. 
     3. We want to persist such kind of data as we may need to deploy a new version of our app with new container and stopping the previous container and running a new container should not lose our data. We need to fetch those data into the app in the container.
     4. Permanent data is read-write data. We want to write it when the app is running.
     5. We store permanent data in container with help of **volume**.