****

**UNIVERSITY OF PETROLEUM & ENERGY STUDIES**

**Dehradun**

**APPLICATION CONTAINERISATION**

**Name: Mansi Saini**

**Course: B. TECH CSE DevOps (2018-22)**

**Roll no.: R171218123**

**Sapid: 500071246**

**Experiment No. 13**

**Prepare and Implement Docker Container Restart Policy**

**So, in this experiment, we are focussing on restart policy in Docker Container. It is performed using Katacoda environment.**

* **Docker considers any containers to exit with a non-zero exit code to have crashed. By default, a crashed container will remain stopped.**
* **Depending on your scenario, restarting a failed process might correct the problem. Docker can automatically retry to launch the Docker a specific number of times before it stops trying.**
* **Finally Docker can always restart a failed container, in this case, Docker will keep trying until the container it is explicitly told to stop.**

**Follow these steps below:**

**1. Launch an instance using the following command:**

****

**2. If you list all the containers, including stopped, you will see the container has crashed.**

****

**3. While the logs will output our message, which in real-life would hopefully indicate information to help us diagnose the issue.**

****

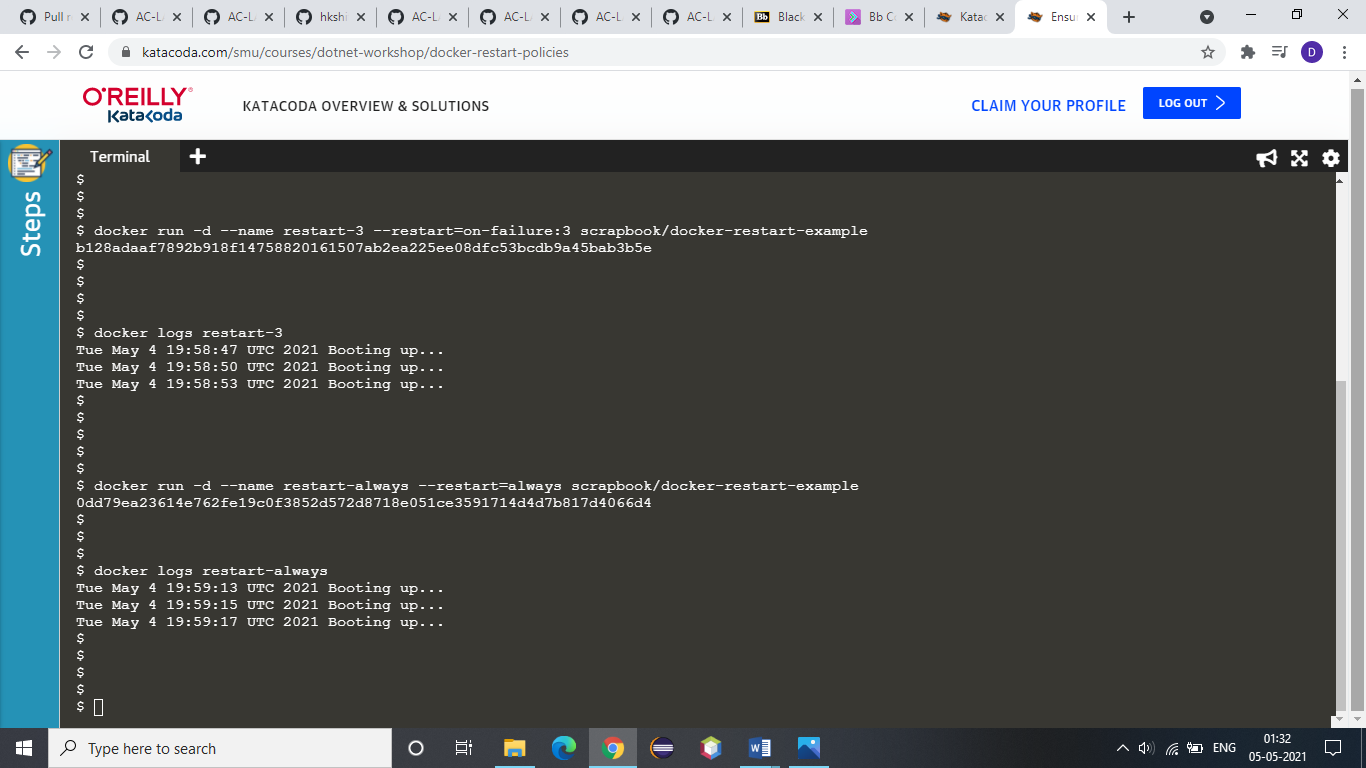
**4. In the example below, Docker will restart the container three times before stopping.**

****

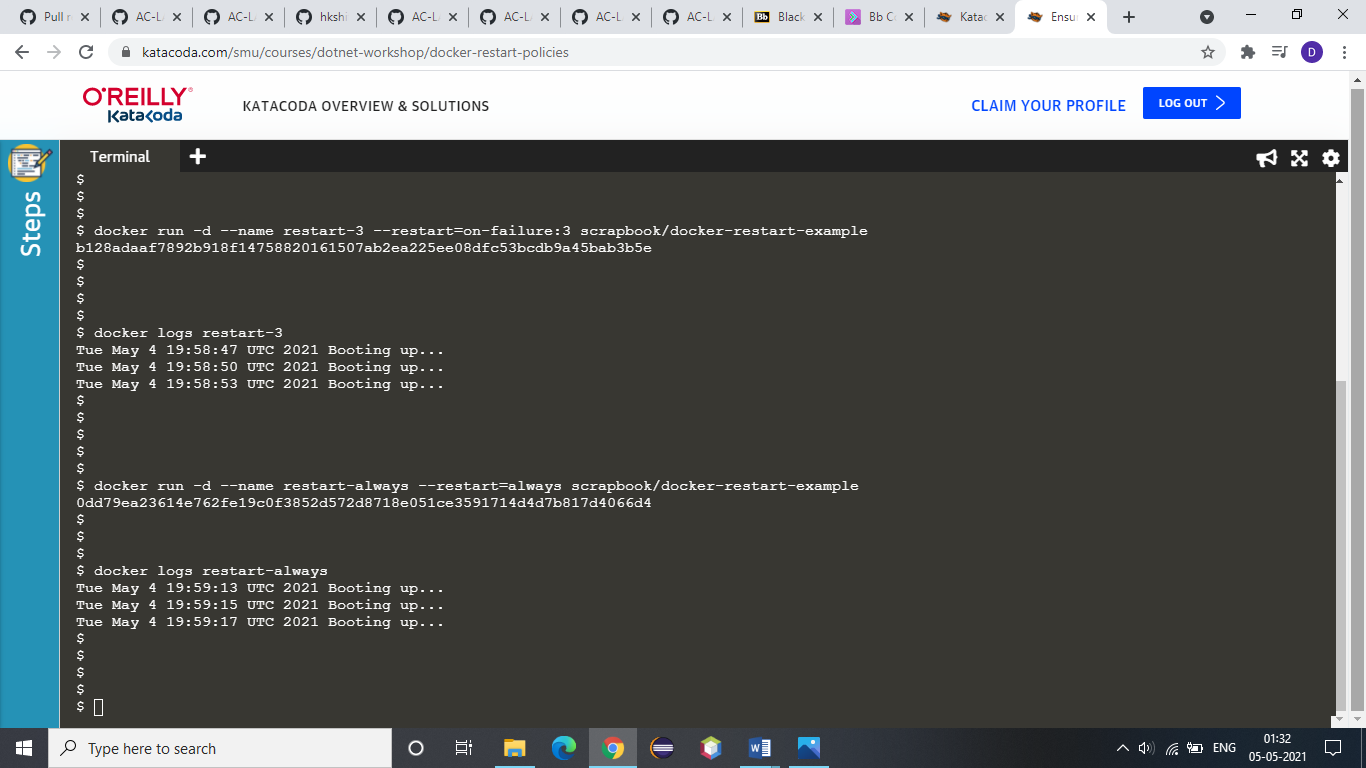
**5. By checking the logs, the above example is clear that it will restart 3 times.**

****

**6. Use the always flag to automatically restart the container when is crashes for example:**

****

**7. You can view the restart attempting via the log as below:**

****