|  |  |
| --- | --- |
| **How to Run Spring Boot Application in Docker** | 1. Create executable jar 2. Create a Spring boot Application 3. Target folder-> right click ->Build using maven config (clean compile install) 4. Once build complete refresh target folder 5. You will find the SNAPSHOT.jar 6. You can run the jar file in cmd in the same location 7. java -jar discovery-server-0.0.1-SNAPSHOT.jar 8. **Create Docker file** 9. Create a file in root directory of boot project say DockerFile.   **FROM openjdk:8-jdk-alpine**  **VOLUME /tmp**  **ADD target/discovery-server-0.0.1-SNAPSHOT.jar app.jar**  **ENTRYPOINT ["java","-jar","/app.jar"]**   1. **Build Docker Image** 2. Go to docker file location 3. Open command prompt 4. docker build -t **discovery-server . (NOTE: name should b in small letter)** 5. docker image ls (list all docker images) 6. **Run the image in docker container**. 7. Run the image 8. **docker run -p4000:8761 disc-server** 9. This command will create new container and start the application. 10. Now how to stop the docker 11. docker ps (all running container) 12. docker container stop container\_id 13. if you want to start service in the same container then 14. docker container start container\_id 15. get all stopped docker ids docker ps -a |
| OS: |  |
| HYPERVISER | e.g: VMWare, Virtual Box |
| Running VM on Server |  |
|  |  |
| Difference: | **VM**:  Reboot time in minutes  Waste of resources  Isolate system  Web hosting companies likely to use VM  Independent House  **Docker**:  Reboot time in seconds  Saves resources  Isolate application  Running multiple services of one web applications  Its like appartments |
| **Docker Images Vs Docker Container** | Docker images are guaranteed to Run in any container. |
| **Run Mongo DB inside docker** | docker pull mongo  docker pull mongo:4.0.4  docker images (List all images)  docker run --name mongodb mongo:4.0.4  docker exec -it mongodb bash (this will start interactive mode..)  to create new DB : use tapan  insert object: db.people.save({firstName:"Tapan",lastName:"Achary"})  db.people.find(): { "\_id" : ObjectId("5e6e9f1daa003a10080470a8"), "firstName" : "Tapan", "lastName" : "Achary" } |
|  |  |
|  |  |