Creating Docker Config:

1. Create Dockerfile:

```
# Alpine Linux with OpenJDK JRE
FROM openjdk:8-jre-alpine
# copy JAR into image
COPY main-app-module-0.0.1-SNAPSHOT.jar /main-app-module-0.0.1-SNAPSHOT.jar
COPY application.qa.yaml /application.qa.yaml
COPY log4j2.yml /log4j2.yml
# run application with this command line
RUN /bin/sh -c "apk add --no-cache bash"
ENTRYPOINT
        ["/usr/bin/java","-Dspring.profiles.active=qa","-Dspring.config.location=/application.qa.yaml","-Dlogging.config=/log4j2.yml","-jar","/main-app-module-0.0.1-SNAPSHOT.jar","DATA_MANAGEMENT",">","nohup.out","2>&1","|","echo","$!",">","data_pid.txt", "&"]
```

2. Run:

```
sudo docker build -t data_management .
```

Please note that this expects main-app-module-0.0.1-SNAPSHOT.jar, application.qa.yaml and log4j2.yml in the same directory as Dockerfile

3. This will create a docker image with name data_management.

4. Create a tag of this image and push this on Docker Registry server using:

```
sudo docker tag data_management docker-registry.nonton.99array.com:5443/data_management
```

```
sudo docker push docker-registry.nonton.99array.com:5443/data_management
```

If prompted for login, use with username and password : docker-admin/3Zj7btENx9hb

```
sudo docker login docker-registry.nonton.99array.com:5443
```

5. Pull Docker image using:

```
sudo docker pull docker-registry.nonton.99array.com:5443/data_management
```

6. To deploy this image, use:

```
sudo docker run -it -p 7073:6063 -d docker-registry.nonton.99array.com:5443/data_management
```

This deploys the container in detached mode.

7. To attach this container to current tty, use:

```
sudo docker ps
sudo docker exec -it <container id> /bin/bash
```

Please note, do not “exit” from the attached bash, as it will kill the container.
To detach from the container, use Ctrl+P and Ctrl+Q .