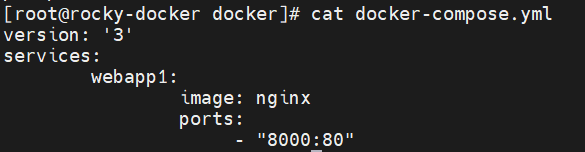
1. **Docker compose is a tool for defining and running multi-container docker applications. E.g. wordpress application. Basic wordpress application contains 2 tiers namely web tier and database tier.**
2. **#docker-compose up (to run docker compose file)**
3. **#docker-compose down**

**This command deletes everything created through docker compose file except volume. If you wish to delete volume as well then use below command.**

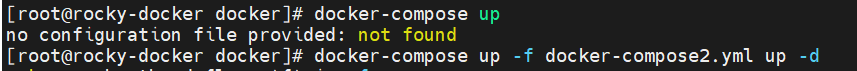
**#docker-compose down –volume**

1. **Creating first docker compose file.**

****

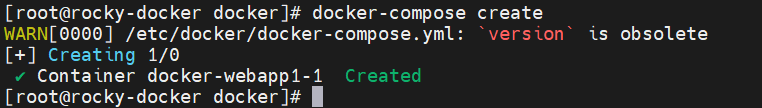
1. **If you have added multiple container creation code in compose file and afterwards if you make any changes in particular container code then docker will not recreate all containers. It will read compose file and recreate the only container where changes are made.**
2. **Docker default checks for docker-compose.yml file. If you wish to give some other names then you need to mention file name along with compose command. Because in project you have to create multiple different compose files.**

**#docker-compose up -f docker-compose2.yml up -d**

****

1. **#docker-compose create**

**This command will create container without dedicated network. On other hand #docker-compose up command creates container with own dedicated network.**

****

1. **Some more basic commands as follow,**

**#docker-compose start**

**#docker-compose stop**

**#docker-compose rm**

**Please note that above 3 commands will not make any network or volume related changes. Because they do not create any dedicated volume or network unlike up/down command.**

1. **Below some more useful commands,**

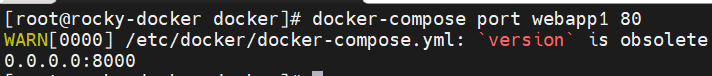
**#docker-compose ps**

**#docker-compose pause**

**#docker-compose unpause**

1. **#docker-compose port webapp1 80**

**This command will tell you port(here e.g 80) of your container is publicly mapped with which port.**

****

1. **#docker-compose logs -f webapp1**

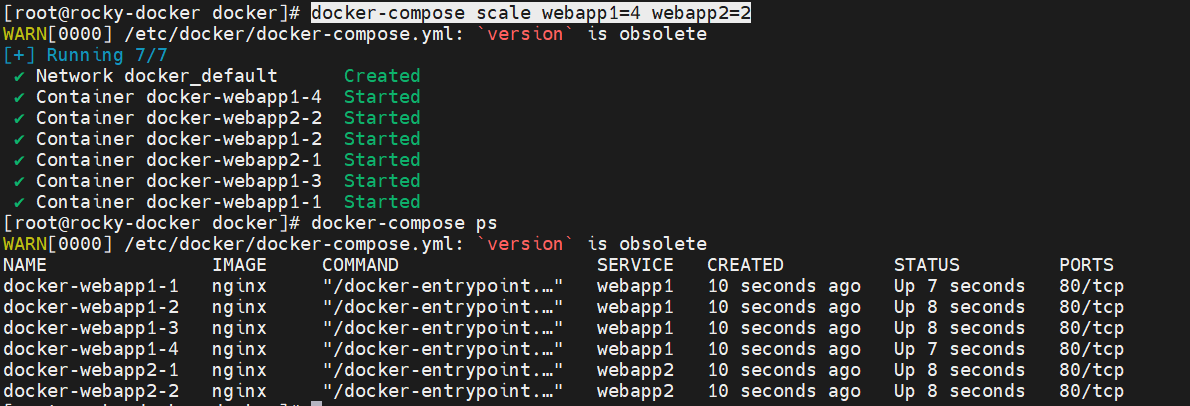
**To check logs of your running container**

1. **#docker-compose exec webapp1 ls**

**To run command inside your running container**

1. **#docker-compose scale webapp1=4 webapp2=2**

**This command will generate/create the number of containers. It is like autoscaling.**

****

1. **When you start working on project by using visual studio app. So in that you are going to create dockerfile and docker-compose yaml file. In dockerfile build details are mentioned. In docker-compose file other details are given like image name, port mapping etc.**
2. **It is best practice to keep .env file in docker project so your code/project become more generic since .env file holds defined variable details which you can call in docker-compose code.**