Docker Document

1. Creating the environment and installing docker and checking it

1 sudo apt-get update

2 sudo apt-get install apt-transport-https ca-certificates curl software-properties-common

3 curl -fsSl https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -

4 sudo apt-key fingerprint 0EBFCD88

5 sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb\_release -cs) stable"

6 sudo apt-get install docker-ce

7 sudo docker info

8 sudo docker container ls -all

2)

44 docker --version

45 which docker

46 sudo groupadd docker

47 sudo usermod -aG docker $USER

48 newgrp docker

49 docker network

50 docker network ls

51 docker inspect eff40aac7078

52 history

53 docker images

54 docker pull busybox

55 docker ps

56 docker ps -a

57 docker run busybox

58 docker ps

59 docker ps -a

60 docker run -d busybox

61 docker ps

62 docker ps -a

63 docker run -d --name server busybox

64 docker run -d --name server1 busybox

65 docker ps -a

66 docker build --help

67 docker run -d --name server busybox sleep 100

68 docker run -d --name server2 busybox sleep 100

69 docker ps

70 docker stop 245280

71 docker ps

72 docker start 245280

73 docker ps

74 docker stop server2

75 docker ps

76 docker ps -a

77 docker rm server2

78 docker rm server

79 docker rm server1

80 docker rm 7a68

81 docker rm c884c85

82 docker ps -a

83 docker rm 58b

84 docker rm 3d8c

85 docker ps -a

86 docker run -d --help

87 docker run -d --name server --help

88 docker run -d --name server --rm busybox ifconfig a

89 docker ps -a

90 docker ps

91 docker ps -a

92 docker run -it --help

93 docker ps

94 docker ps -a

95 docker rm -aq

96 docker rm -q

97 docker rm -h

98 cd

99 docker run -e MYSQL\_ROOT\_PASSWORD=admin -e MYSQL\_USER=wordpress -e MYSQL\_PASSWORD=wordpress -e MYSQL\_DATABASE=wp\_database -v wp\_database:/var/lib/mysql --name wordpressdb -d mariadb\

100 docker run -e MYSQL\_ROOT\_PASSWORD=admin -e MYSQL\_USER=wordpress -e MYSQL\_PASSWORD=wordpress -e MYSQL\_DATABASE=wp\_database -v wp\_database:/var/lib/mysql --name wordpressdb -d mariadb

101 docker ps

102 docker inspect 55e43a057263

103 docker run -e WORDPRESS\_DB\_USER=wordpress -e WORDPRESS\_DB\_PASSWORD=wordpress -e WORDPRESS\_DB\_NAME=wp\_database -e WORDPRESS\_DB\_HOST=172.17.0.2 -p 8080:80 -v wp\_data:/var/www/html --name wordpressapp -d wordpress

104 docker ps

105 docker run -e WORDPRESS\_DB\_USER=wordpress -e WORDPRESS\_DB\_PASSWORD=wordpress -e WORDPRESS\_DB\_NAME=wp\_database -e WORDPRESS\_DB\_HOST=172.17.0.2 -p 8080:80 -v wp\_data:/var/www/html --name wordpressapp -d wordpress

106 docker ps

107 docker exec -it 6db /bin/bash

108 docker inspect 6db

109 docker ps

110 docker -rm 6db

\*\*\*\*\*----------------------------------Busybox-------------------------------------------

118 docker pull busybox

119 docker run -d --name server busybox

120 docker run -d --name server busybox sh -c "while true; sleep 5; done"

121 docker run -d --name server1 busybox sh -c "while true; sleep 5; done"

122 docker ps

123 docker ps -a

124 docker ps -a -q

125 docker ps -a

126 docker rm $(docker ps -a -q)

127 docker ps -a

128 docker run -d --name server1 --rm busybox ifconfig a

129 docker ps -a

############---------------- Closed Networking Model------------------------------

130 docker run -it --rm --net none busybox ifconfig -a

131 docker ps -a

#############----------------- Bridged Networking Model: DEFAULT MODE------------------------------

132 docker run -it --rm --net bridge busybox ifconfig -a

133 docker run -d -p 80 --name webserver httpd

134 docker stop webserve

135 docker stop webserver

136 docker rm webserver

###########--------------------- Open Networking Model--------------------------

137 docker run -d -p 8080:80 --name webserver httpd

138 ip add

139 docker run -d -p 127.0.0.1:8080:80 --name webserver httpd

140 docker run -d -p 127.0.0.1:8080:80 --name webserver1 httpd

141 docker rm $(docker ps -a -q)

142 docker ps -a

143 docker -rm cceb

144 docker rm cceb

145 docker ps -a

146 docker stop cceb

147\* docker rm

148 docker run -d -p 127.0.0.1:8080:80 --name webserver httpd

149 docker stop webserver

150 docker rm webserver

151 docker ps -a

152 docker run -d --name webserver --net host httpd

153 docker stop webserver

154 docker rm webserver

#######----------- Bind Mounts-------------

156\* docker run -d --name webserver -p 8080:80 -v "$(pwd)":/usr/local/apache2/htdocs http

157 ls -ld

158 pwd

159 cd .

160 ls

161 cd /opt/

\*\*\*\*\*\*\*\*Closed Networking Model\*\*\*\*\*\*\*\*\*\*\*\*\*\

37 docker run -it --rm --net none busybox ifconfig -a\

\

38 \*\*\*\*\*\*\*\*Bridged Networking Model: DEFAULT MODE\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\

39 docker run -it --rm none busybox ifconfig -a\

40 docker run -it --rm busybox ifconfig -a\

41 docker run -it --rm --net bridge busybox ifconfig -a\

\

42 \*\*\*\*\*\*\*\*\*\*\*\*\*Open Networking Model\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\

43 docker run -d -p 80 --name webserver httpd\

44 docker ps \

45 docker stop webserver \

46 docker rm webserver \

47 docker run -d -p 8080:80 --name webserver httpd\

48 docker ps \

49 ip addr \

50 clear \

51 docker run -d -p 127.0.0.1:8080:80 --name webserver httpd\

52 docker rm webserver \

53 docker stop webserver \

54 docker rm webserver \

55 docker run -d -p 127.0.0.1:8080:80 --name webserver httpd\

56 docker stop webserver \

57 docker rm webserver \

58 docker run -d -p 127.0.0.1::80 --name webserver httpd\

59 docker ps \

60 ###### Open Model ##########\

61 docker stop webserver \

62 docker rm webserver \

63 docker run -d --name webserver --net host httpd\

64 docker ps \

65 docker stop webserver \

66 docker rm webserver \

67 docker ps \

68 docker run -d --name webserver --net host httpd\

69 ifconfig \

70 docker stop webserver \

71 docker rm webserver \

72 docker run -d --name webserver -p 8080:80 -v "$(pwd)":/usr/local/apache2/htdocs httpd\

73 docker pc \

74 docker ps\

75 vim test.html\

76 vi test.html\

77 nano test.html\

78 cat test.html\

86 docker stop webserver \

87 docker rm webserver \

88 mv /root/test.html .\

89 docker run -d --name webserver -p 8080:80 -v "$(pwd)":/usr/local/apache2/htdocs httpd\

90 cat test.html\

91\

92 #####------------------------ Above config is for Bind Mounts\

93 docker stop webserver && docker rm websrver\

94 docker stop webserver && docker rm webserver\

95 docker volume create web-data-vol\

96 docker volume inspect web-data-vol\

97 cd /var/lib/docker/volumes/web-data-vol/\_data\

\

98 ####### Named Volume\

99 docker run -d --name webserver -p 8080:80 -v web-data-vol:/usr/local/apache2/htdocs httpd\

100 ls -l\

101 cat index.html \

102 mv /opt/test.html .\

\

103 ########## TMPFS Mounts\

104 clear\

105 cd\

106 docker run -ild --name secureserver --tmpfs /tmp busybox\

108 su \

109 history \

110 exit \

111 history \

112 docker run -itd --name secureserver --tmpfs /tmp busybox\

113 docker stop secureserver && docker rm secureserver\

114 docker run -itd --name secureserver --tmpfs /tmp busybox\

115 docker attach secureserver \

116 docker ps \

117 docker start secureserver \

118 docker attach secureserver \

119 docker rm secureserver \

120 docker run -itd --name secureserver busybox\

121 docker attach secureserver \

122 docker start secureserver \

123 docker attach secureserver \

124 history \

125 clear \

\

\

\

\

\

\

126 #########LAB LAMP Stack###\

127 docker run -e MYSQL\_ROOT\_PASSWORD=admin -e \

128 MYSQL\_USER=wordpress -e MYSQL\_PASSWORD=wordpress -e \

129 MYSQL\_DATABASE=wp\_database -v wp\_database:/var/lib/mysql --name \

130 docker run -e MYSQL\_ROOT\_PASSWORD=admin -e MYSQL\_USER=wordpress -e MYSQL\_PASSWORD=wordpress -e MYSQL\_DATABASE=wp\_database -v wp\_database:/var/lib/mysql --name wordpressdb -d mariadb\

131 apt-get install mysql-client\

132 docker ps \

133 mysql -u webpress -pwebpress -h\

134 docker inspect wordpressdb \

135 mysql -u webpress -pwebpress -h 172.17.0.3 \

136 mysql -u webpress -pwebpress -h 172.17.0.3 Asudo \

137 sudo mysql -u webpress -pwebpress -h 172.17.0.3 \

138 sudo mysql -u wordpress -pwordpress -h 172.17.0.3 \

139 docker run -e WORDPRESS\_DB\_USER=wordpress -e WORDPRESS\_DB\_PASSWORD=wordpress -e W

ORDPRESS\_DB\_NAME=wp\_database -e WORDPRESS\_DB\_HOST=172.17.0.3 -p 8080:80 -v wp\_data:/var/www/html --name wordpressapp -d wordpress\

140 docker stop wordpressdb \

141 docker rm wordpressdb \

142 docker rm wordpressapp \

143 docker run -e WORDPRESS\_DB\_USER=wordpress -e WORDPRESS\_DB\_PASSWORD=wordpress -e WORDPRESS\_DB\_NAME=wp\_database -e WORDPRESS\_DB\_HOST=172.17.0.3 -p 8080:80 -v wp\_data:/var/www/html --name wordpressapp -d wordpress\

144 docker stop webserver \

145 docker rm webserver \

146 docker run -e WORDPRESS\_DB\_USER=wordpress -e WORDPRESS\_DB\_PASSWORD=wordpress -e WORDPRESS\_DB\_NAME=wp\_database -e WORDPRESS\_DB\_HOST=172.17.0.3 -p 8080:80 -v wp\_data:/var/www/html --name wordpressapp -d wordpress\

147 docker rm wordpressapp \

148 docker run -e WORDPRESS\_DB\_USER=wordpress -e WORDPRESS\_DB\_PASSWORD=wordpress -e WORDPRESS\_DB\_NAME=wp\_database -e WORDPRESS\_DB\_HOST=172.17.0.3 -p 8080:80 -v wp\_data:/var/www/html --name wordpressapp -d wordpress\

149 docker ps \

150 docker run -e MYSQL\_ROOT\_PASSWORD=admin -e MYSQL\_USER=wordpress -e MYSQL\_PASSWORD=wordpress -e MYSQL\_DATABASE=wp\_database -v wp\_database:/var/lib/mysql --name wordpressdb -d mariadb\

151 docker ps \

152 sudo mysql -u wordpress -pwordpress -h 172.17.0.3 \

153 exit \

154 history \

155 docker ps \

156 docker run -e WORDPRESS\_DB\_USER=wordpress -e WORDPRESS\_DB\_PASSWORD=wordpress -e WORDPRESS\_DB\_NAME=wp\_database -e WORDPRESS\_DB\_HOST=172.17.0.3 -p 8080:80 -v wp\_data:/var/www/html --name wordpressapp -d wordpress\

157 docker ps \

158 docker stop wordpressdb \

159 docker rm wordpressdb \

160 docker stop webserver \

161 docker ps \

\

\

\

\

\

\

162 ####### LAMP start $$$$$$\

163 clear \

164 history \

165 docker run -e MYSQL\_ROOT\_PASSWORD=admin -e MYSQL\_USER=wordpress -e MYSQL\_PASSWORD=wordpress -e MYSQL\_DATABASE=wp\_database -v wp\_database:/var/lib/mysql --name wordpressdb -d mariadb\

166 apt-get install mysql-client\

167 docker inspect wordpressdb\

168 sudo mysql -u wordpress -pwordpress -h 172.17.0.2\

169 docker run -e WORDPRESS\_DB\_USER=wordpress -e WORDPRESS\_DB\_PASSWORD=wordpress -e WORDPRESS\_DB\_NAME=wp\_database -e WORDPRESS\_DB\_HOST=172.17.0.2 -p 8080:80 -v wp\_data:/var/www/html --name wordpressapp -d wordpress\

170 docker rm wordpressapp \

171 docker run -e WORDPRESS\_DB\_USER=wordpress -e WORDPRESS\_DB\_PASSWORD=wordpress -e WORDPRESS\_DB\_NAME=wp\_database -e WORDPRESS\_DB\_HOST=172.17.0.2 -p 8080:80 -v wp\_data:/var/www/html --name wordpressapp -d wordpress\

172 docker ps \

173 docker volume inspect wb\_daa\

174 docker volume inspect wb\_da\

175 docker volume inspect wb\_daTA\

176 docker volume inspect wb\_data\

177 docker volume inspect wp\_dat\

178 docker volume inspect wp\_data\

179 cd /var/lib/docker/volumes/wp\_data/\_data\

180 ls -l \

181 hisory \

182 history\

\

}