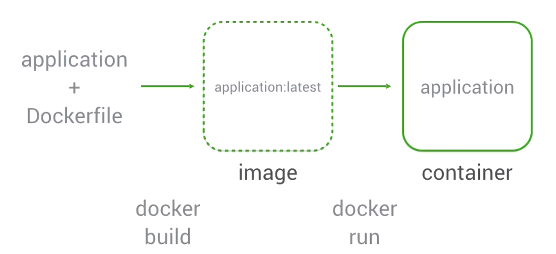
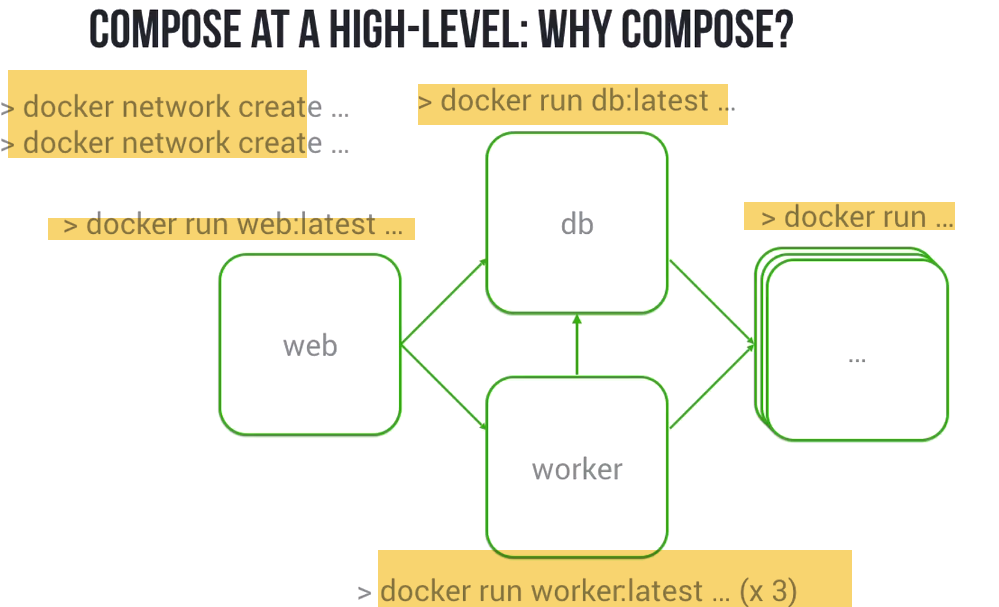
Docker-compose🡪Orchestrating (Build) multi container environments



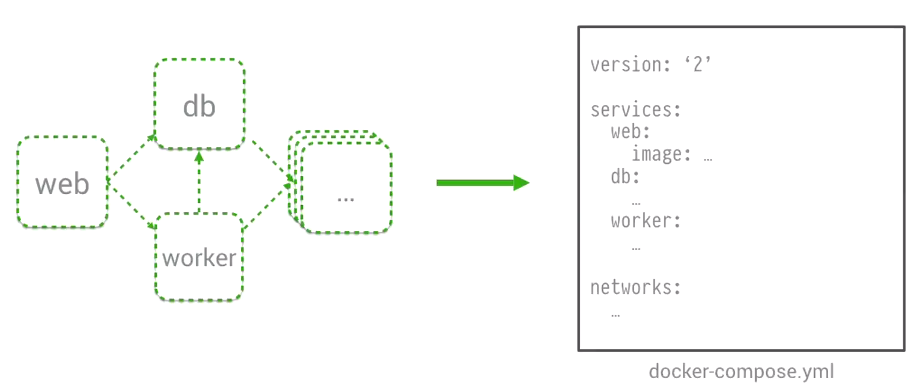
Why Docker-compose?

-We will be needing to run many commands to start env.

- Tough to run these commands reproducibly on own.



What Compose does iS::It captures network, volumes,Config. of services etc..you have in docker env..using a single docker-compose.yaml file.



Then rather than running forcibly running each and every command, single command up will bring setof comnatiners, n/w,vlumes etc(environment).

**On Dev.Env:**

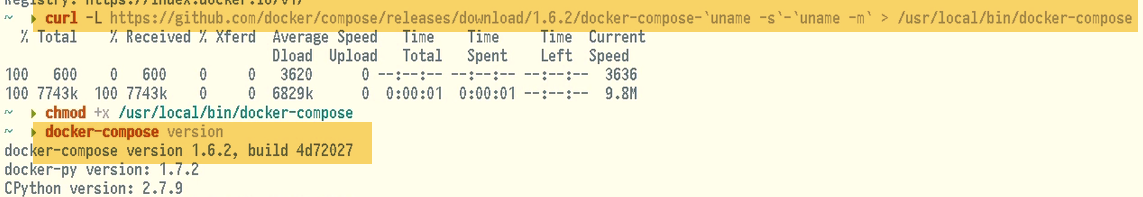
--Brings version of your system(devt env. Or own box) similar to prod.

--Allow to interact, Make changes and deploy it

**On QA:**

🡪Facilitates automated test env.Since with single command, we can launch environmnet..Facilitares to launch any version of env..,and tear everything down.

--Allows to test in a env.., really seen on Prod env.



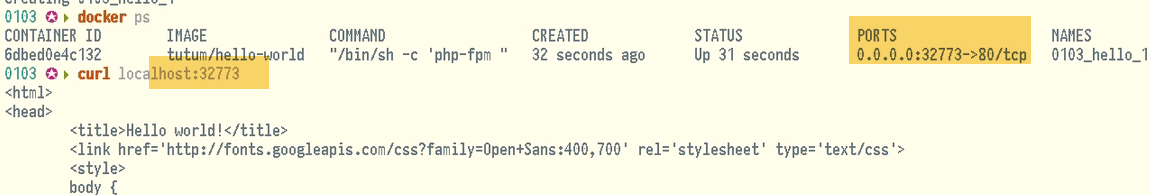
**EX:** 

YAML🡪Key-value, array

Services: Container names, and config.

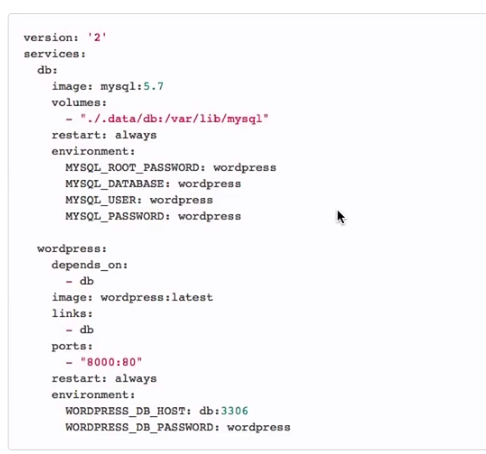
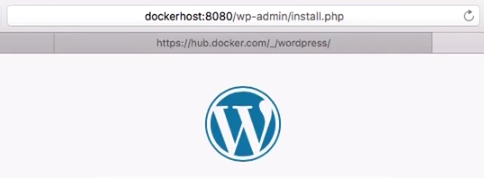
We have container hello, using image tum…, and expose port 80

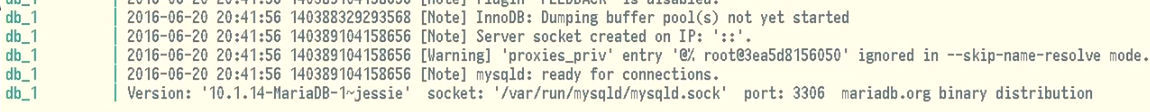
Pulls image, Creates n/w, Run hello-world container

**Test:**

**EX:-2:** 

We have Application Container, accepting connections on port 8080.DB container that stores info. of that Wordpress container.



Version: 2 Use docker new features

**Here error, saying DB password ENV. Var is missing. The way wordpress image is written is taking advantage of features in compose version-1 , and don’t ask for password.. Not present in verion-2.**

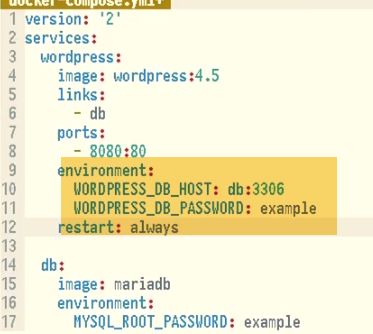
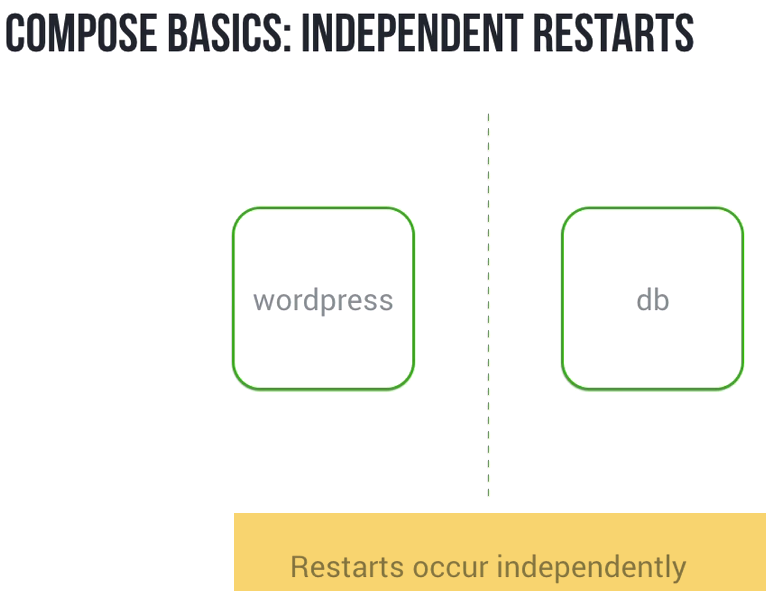
**Version-1 allows some implicit env. Variable .**

**-🡪 The container DB is linked as MYSQL is picking up an implicit env. Var, and setting wordpress DB password**  

**Note: Pick specific image tag,rather than latest**

**DB to MYSQL link is setup in way it was..,aliasing DB container to Mysql.It’s trick to take advantag o some implicit variables.No need tof alias..Just link as DB.**

**But, we need to tell wordpress, how to start, where & how to find db,what crdentials it can use**

**Restart: If ever,containerneeds to stop---restart**   

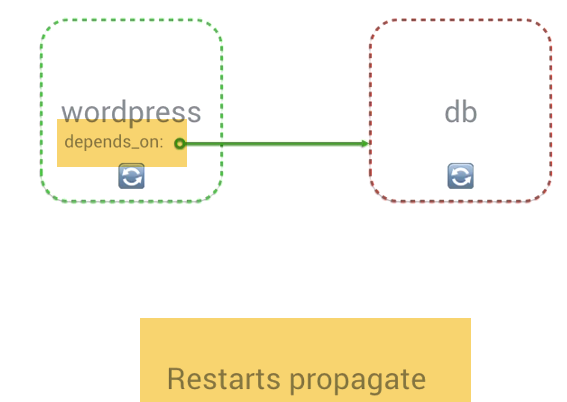
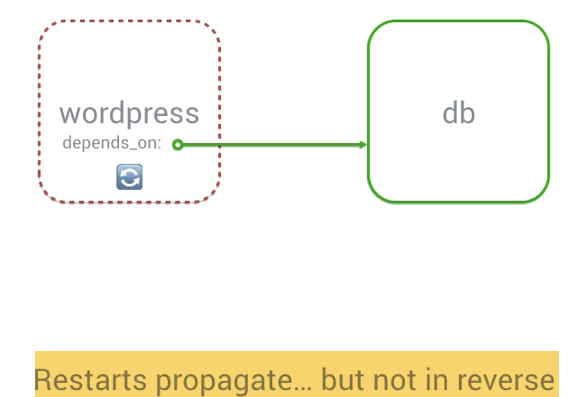
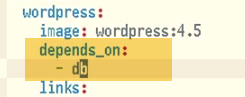
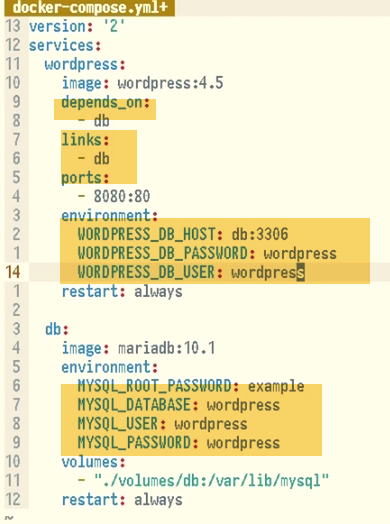
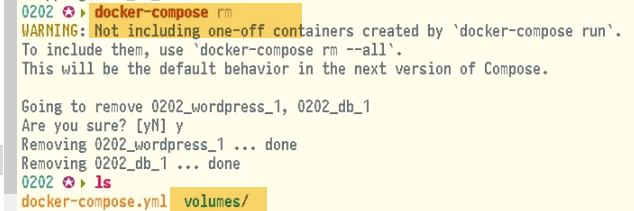
If DB goes down in this env,restart policy will cause to restart it..Independently of wordpress

If DB disappeers, gets replaced.So no hierarchy…Just use depends\_on

Depends\_on setup hierarchial dependency b/w containers.Wordpress container depends on DB container, which means restart is not independent,dependant on each other.

**IF DB goes down,restart will actually propate from DB upto wordpress.If DB is down,and if restarted..Wordpress allow will be restarted,even if it is running fine.Compose will take it down, and restrt after DB is restarted..Thus both will be up again.However, restart doent propagate other way.**

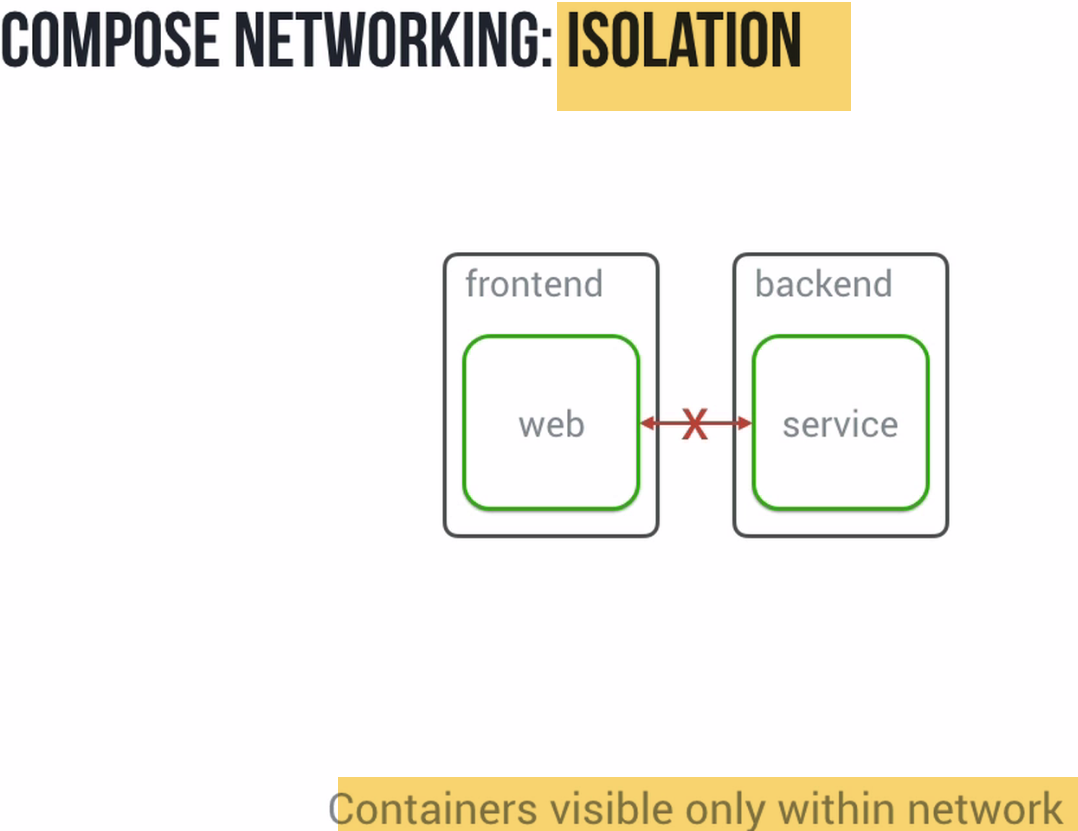
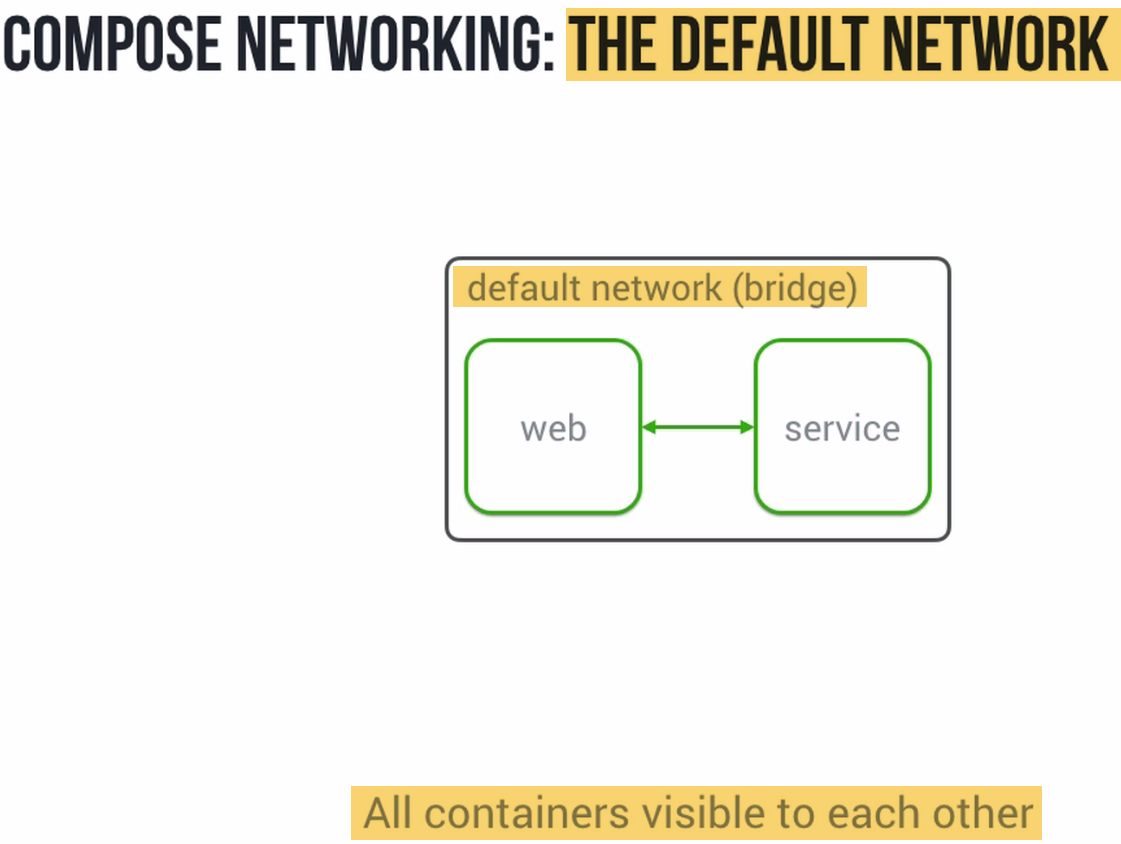
**If wordpress fails,it depends on DB.It is not dependent on itself.That restart wont propagate back.**

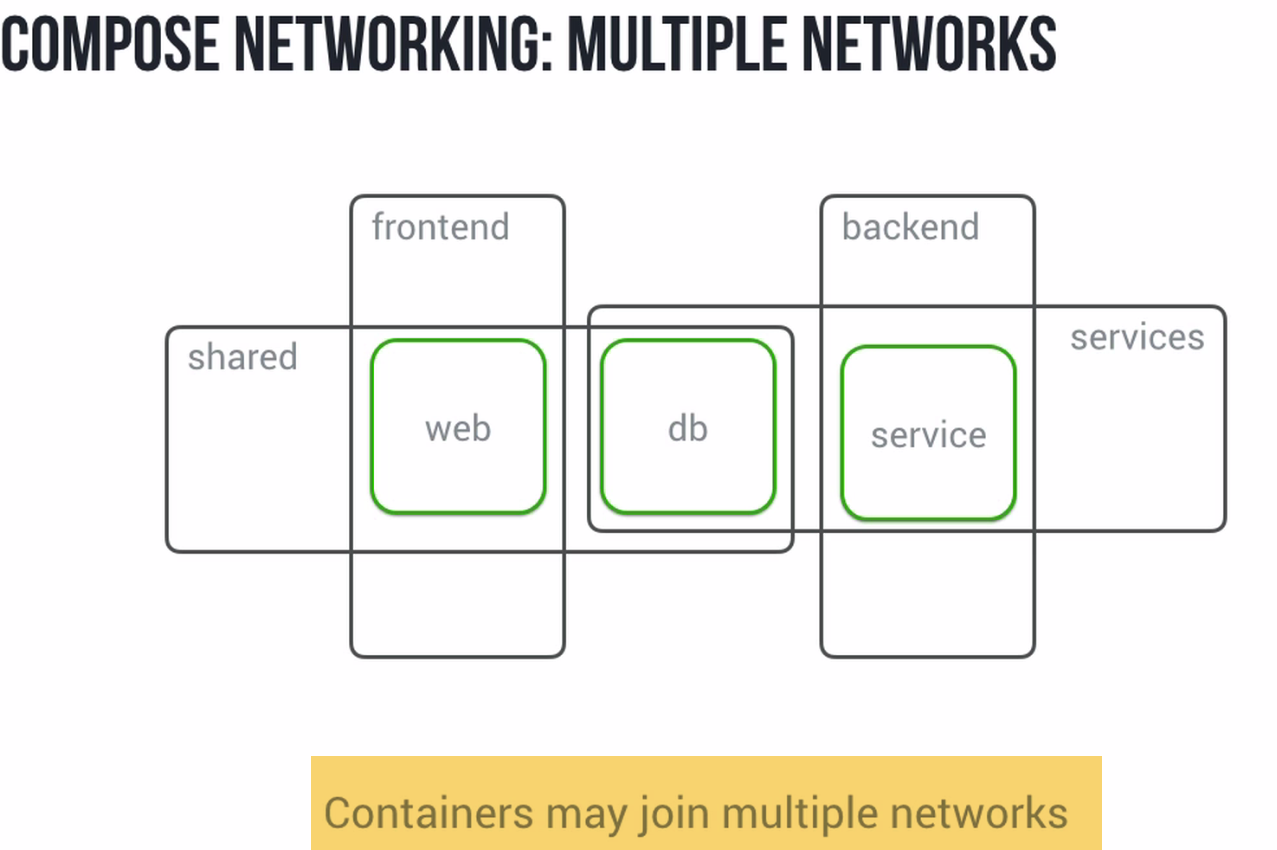
    

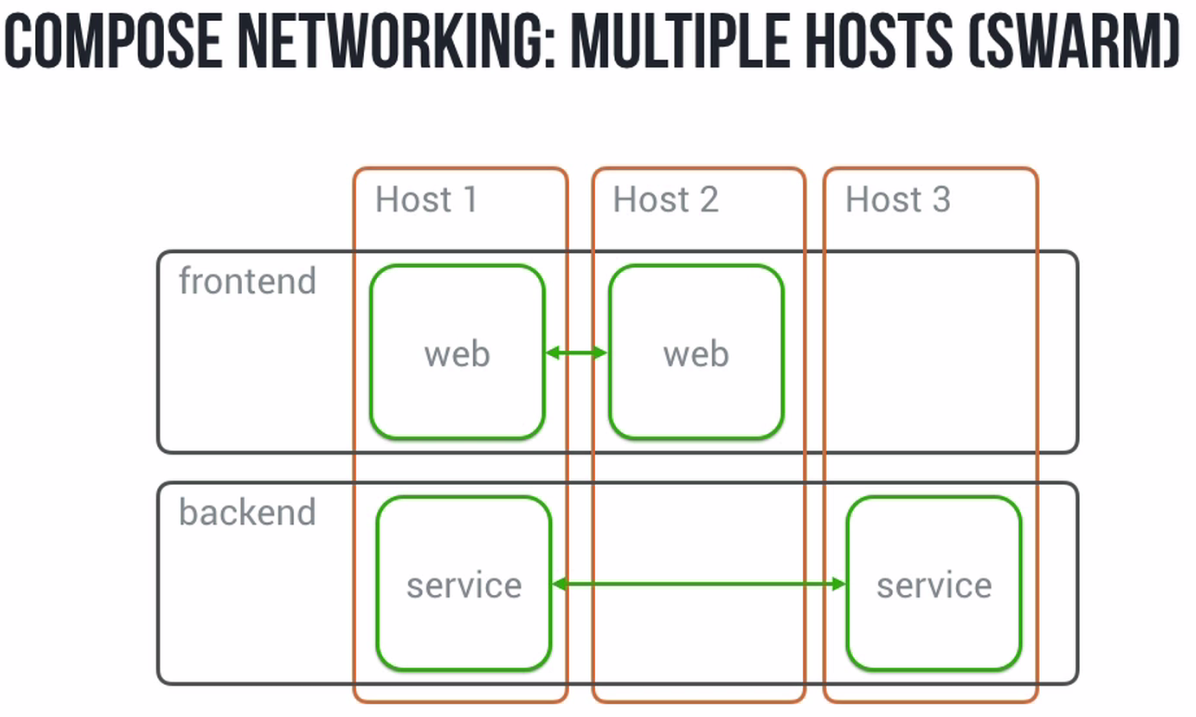
**Even after rm..,volumes are persistent**

🡪DO Docker-compose again, we can use volumes..

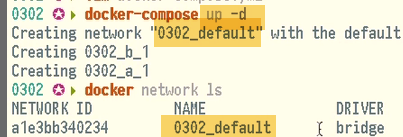
# **Ntworking Overview**

---Cannot communicate with containers on other n/w.

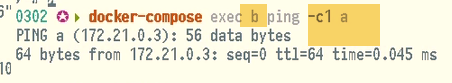
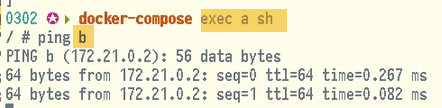


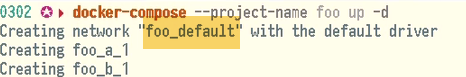


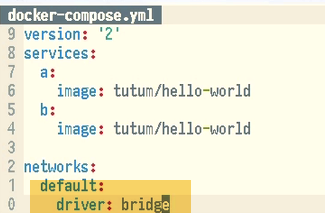
# **Default Network**

We have a service, b service

-🡪Launch shel inside conainer: 

🡪Creating new N/W

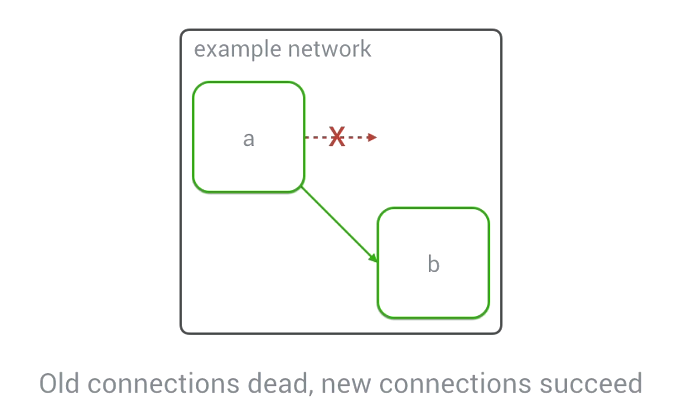
🡪Change to default n/w: 

17 Hw Updates Affect Networking

From time to time, we wwant to make updates to our env..

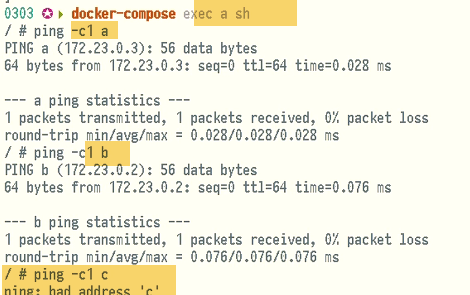
Ex:1.)

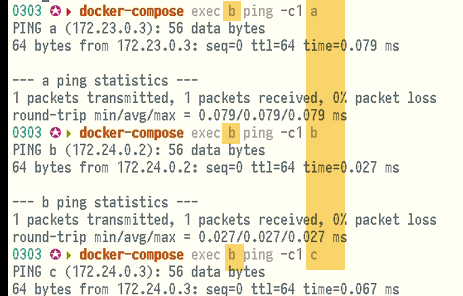
**A is communicating with b.., and has open connection**

**2.) If we make change to container-B and it is subsequently replaced. We will find old connection on A still open, and unable to connect to old container-B.** 

**However, if we attempt to make new connection from a to b will immediately see the new container..**

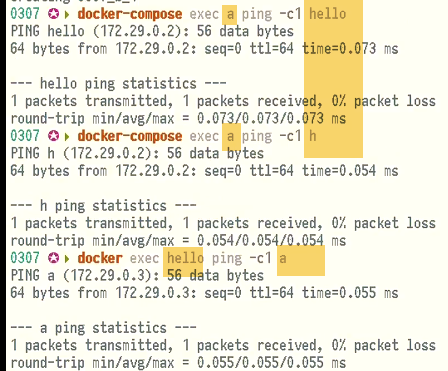
# **Isolating Containers**--Front end—b,a --Backend-b,c

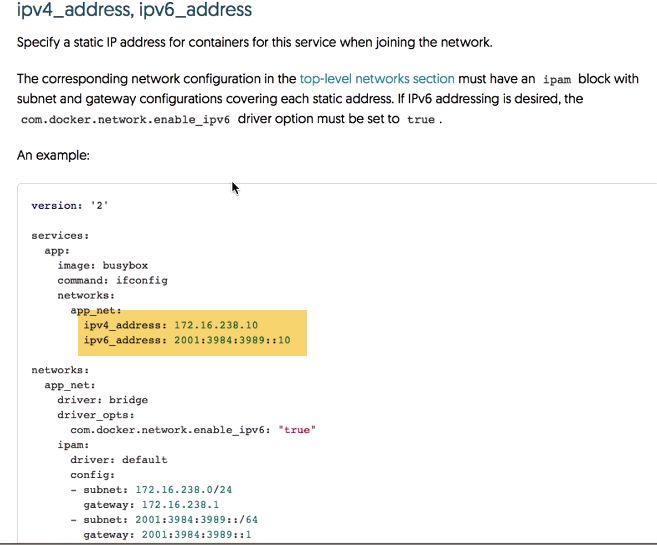
-🡪 a, b can communicate..c not visible

-🡪 a,c visible to b

# 19 Usng External Networks

1. Rather than specifying driver option, specifying external=true which indicates we are going to use external network
2. If we want to add one of container-A to superbridge n/w. It should be able to see hello-world container.Alos, if we want we can also provide our own external\_link that we make our containers available as.
3. a.) Here we specify hello-world container, available as h

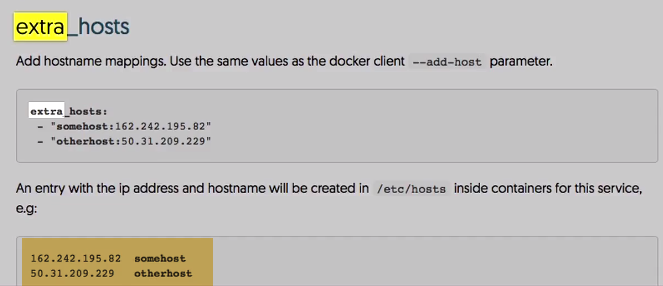


🡪We can make further customizations likes setting ipv4,ipv6 address.

-🡪2 ways of : Gaining access othr external hosts,from containers inside ur environment

Ex: may exist on local n/w or you have custom names you use.

In many cases, you may have only DNS server working in your org. In this you can configure DNS option on your services to set custom n/w that reflect you own n/w configuration.

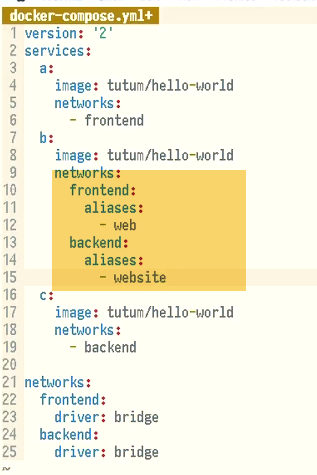
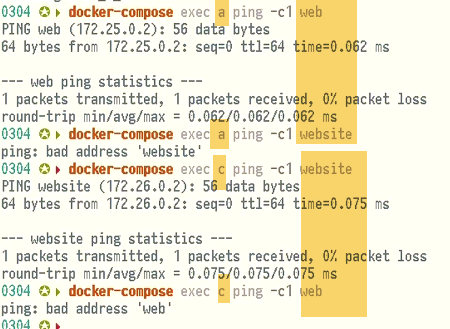
🡪Finally, extra\_hosts option: we can provide hostname alias, in similar fashion to --add-host file to each of containers

# **20 Aliases & Container Names**

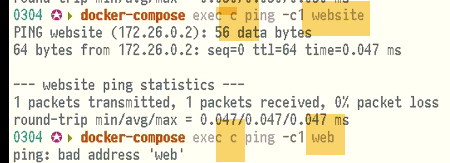
Aliases are alternate hostname to services on n/w.

We can alias d service as web on frontend, and website on backend .

1. From this we need to change list of n/w to set of ..
2. And then we provide list of alias.., under which we want container to be available to in that n/w

(before)(after) a,website doent work..As it is not alias for frontend n/w.

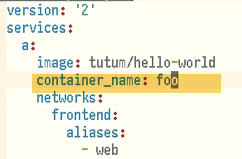
**Alias can also be shared b/w multiple conatiners..**



**C container is able to ping website still. .With web alias, it should point to instance of b in some cases. However, it don’t actually work.**



**From b, c works for web**

**🡪Container names**

**Add container-name key, and add a value. It will give container hostname.**

**Since container names should be unique, specifying this does take a global spot**

