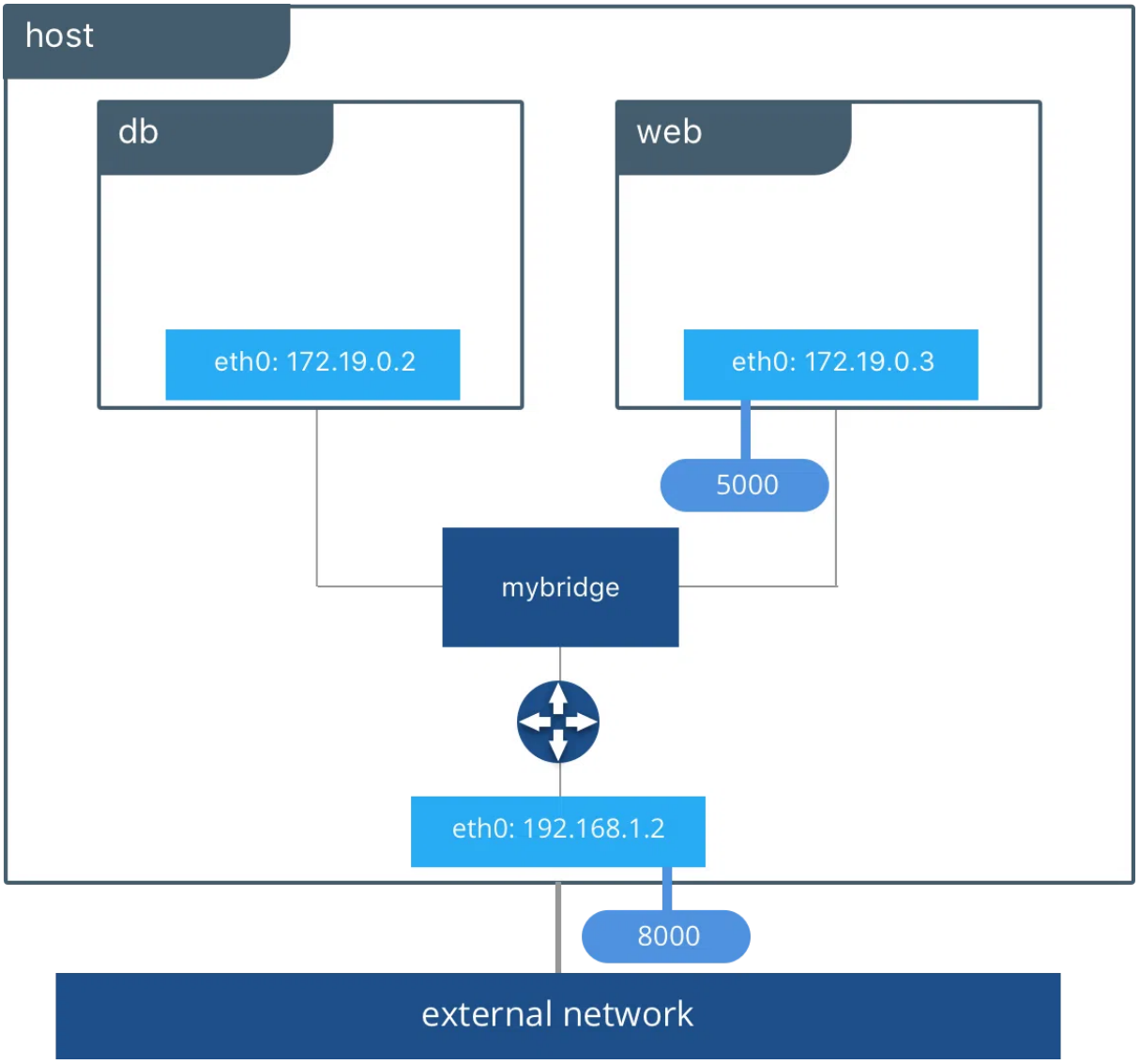
Docker

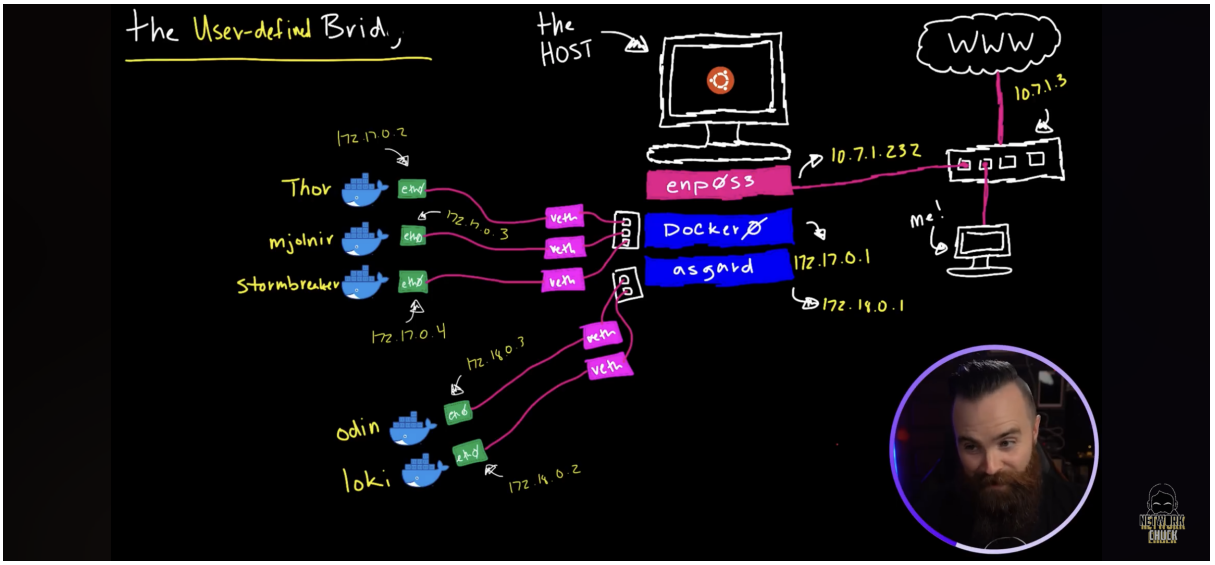
Notes by Sparsh Verma

- A container is a standalone executable package of a software that includes everything required to run that software application, be it the database, runtime configurations, etc.
- Docker is an open platform that provides us the ability to run our applications in an isolated env called containers.
- For eg. There is an app that requires MySql, Redis. In order to run it one has to install all these dependencies in their machine. Here is where docker comes into the play. A docker container running on the docker engine will set up all these configurations in a virtual env into which the app can run.
- A **docker image** is the actual package, the software application along with its configurations that can be shared from one machine to another. A **container** is a running env of an image.
- A OS has 2 major layers, The **Kernel** that interacts with the hardware and the **applications layer** that interacts with the Kernel.

- In VM, the Virtual Machine image contains its own Application layer and Kernel and directly connects with the H/W.
- Docker virtualizes at the Applications layer level. All the containers share the Kernal of the host OS.
- `docker run -d <image_name>`, runs the container in detach mode.
- `docker ps -a`, all containers
- `docker ps`, all running containers
- `docker start <container_id>`, runs a paused/stalled container
- `docker run -p<HOST_PORT>:<CONTAINER_PORT>`, port binding
- to get into the terminal of a container, `docker exec -it <CONTAINER_ID/NAME> /bin/bash` (it stands for interactive terminal)
- The Docker network is a virtual network created by Docker to enable communication between containers. If two containers are running on the same host they can communicate with each other without the need for ports to be exposed to the host machine.
- If no network is specified, the containers are connected to the default **bridge** network.
- We can create our own network in docker, `sudo docker network create --driver <driver-name> <network-name>`
- Containers on the default bridge network can only access each other by IP addresses.
- 2 containers in the same user-defined docker bridge network can connect to each other using their names only.

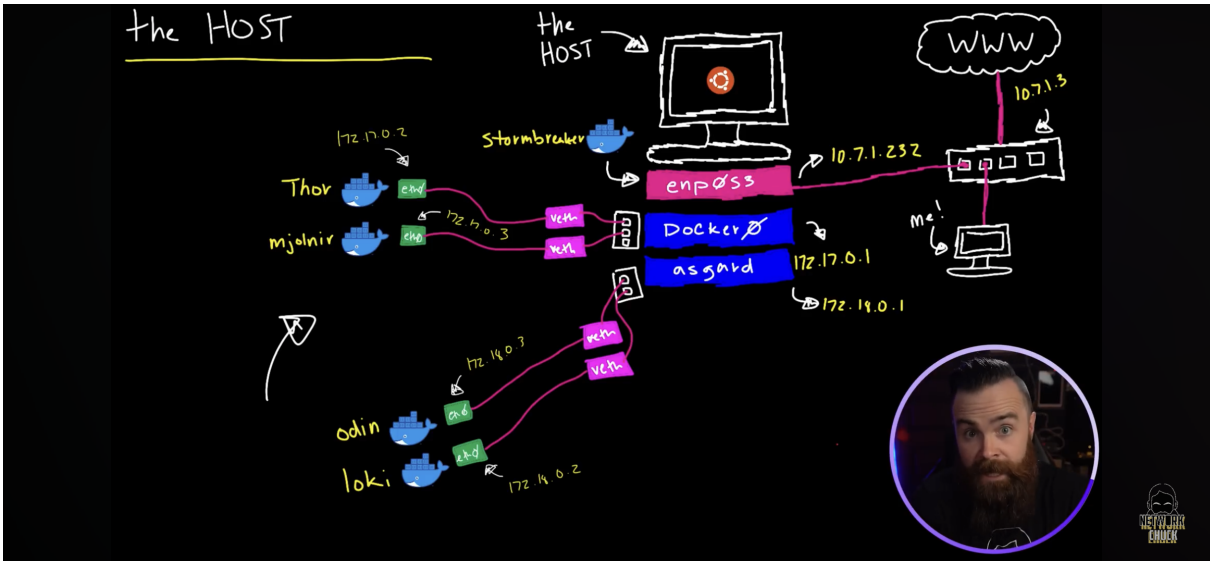


Bridge network mybridge



Odin and Loki are in user-defined bridge network, Asgard

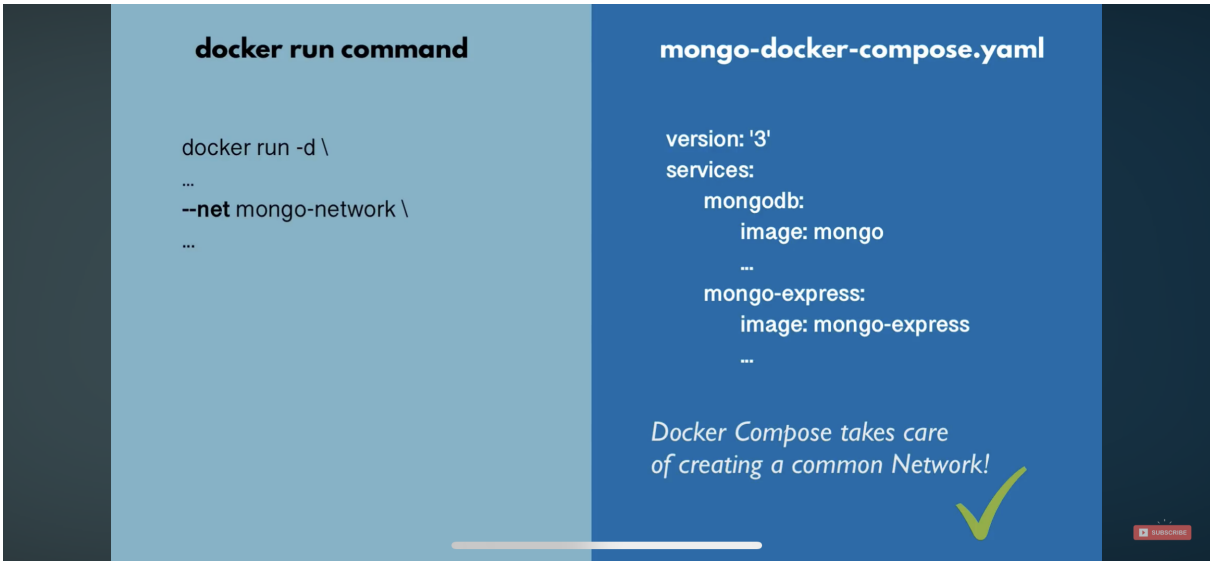
- **Host** network driver allows the container to sit at host level. We don't have to expose the ports too.



Stormbreaker is a host network container

1. **MACVLAN** where each container gets its own MAC address and IP address.
2. **IPVLAN L2** - containers share the MAC address of the host but have their own IP Address.
3. **IPVLAN L3** - containers connect to the host like it is a router.

Docker compose file creates a network for its services to interact.



Docker volumes are stored in the physical/host machine and on container startup, they are copied into them.