Host

Processes
- PID 1
- PID 2
- ⋮
- PID n
- Docker Daemon

* First thing to notice: Docker Daemon runs as a process just like any other process on the host. THEREFORE, it can bind itself to ports on the host!

* AND, most commonly, the Docker Daemon is going to bind itself to whatever ports the containers running on the host are listening on.
* THEREFORE, the docker daemon process seems to act as a router that sits between the host and all of the different docker networks on that host!

Docker Daemon

- Docker Network 1
- Docker Network 2
- ⋮
- Docker Network n

* you can imagine each docker network as sitting behind its own little router w its own IP Address. THEN, just like in subnetting, the "docker daemon router" will use "NAT rules" to forward requests to whichever router (IP) is associated w that service port.
* By default, each container is run in its own isolated subnet == docker bridge network. You can imagine this means that it sits behind its very own router!

Docker Network

- container 1
- container 2
- ⋮
- container n

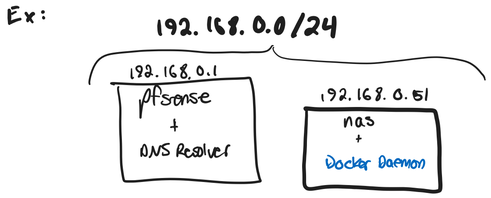
* Wtn each docker network could be any # of containers, just like there could be any # of devices on your subnet at home!
* By default, each container is isolated in its own docker subnet, but containers can be added to the same docker network (subnet) if desired — which is most useful when you want one container to easily talk to the other!
  - In this config, the subnetwork's router will automatically create DNS resolving rules to resolve container names to their respective addresses to make referencing each other more seamless!

* **Big note:** these docker subnets that I'm referencing aren't some magical docker construct! — NO —. Run ip addr to see all of the different virtual networks that docker creates!!

The docker daemon doesn't magically hide these away — it just manages them, basically!

And that's really all the docker Daemon/engine itself is: A tool for managing the containers in a container runtime and how data is routed between those containers!

Ex:



* Docker daemon binds to host ports on behalf of the container port binds!

* Docker Daemon mimics NAT as it forwards requests along to whichever Docker network IP is mapped to the transport layer service in the request!!