Create a client and server podman containers, and figure our how to run them as rootless,rootless

* Running **podman** as an unprivileged user (rootless), with processes in the container running as non-root (also known as rootless as a non-root user)

podman run -it -u sync localhost/nodejs-client sh

podman **unshare** cat /proc/self/uid\_map

non-root container storage

ls ~/.local/share/containers/storage/

# Problems

Network connect is only supported by networks created by a bridge network.

A rootless container has no ability to access a port less than 1024

# Topics

* Podman Client Server in a shared network
* Slirp4netns
* Ipv6
* Running in a pod
* Networks

By default podman creates a **bridged** network in rootless mode.

When you are rootless the slirp4netns is automatically used.

Use -u sync to be a rootless user

* Unshare

podman unshare – rootless-netns

# Facts

Non root containers are at $HOME/.local/share/containers/storage

**podman system reset**

# Experiments

podman create network server

podman build -t nodejs-client .

podman build -t nodejs-server .

podman run --name server -u sync --network server -p 3000:3000 -d nodejs-server

podman run --rm -u sync --network server nodejs-client

Response: Hello World!ls

# Experiment 2

podman create network **server**

podman create --name server -u sync --network **server** -p 3000:3000 nodejs-server

podman create --name client -u sync --network **server** nodejs-client

services:

server:

image: localhost/nodejs-server:latest

container\_name: nodejs-server

restart: always

ports:

- '3000:3000'

networks:

- server

client:

image: localhost/nodejs-client:latest

container\_name: nodejs-client

restart: always

depends on:

- server

networks:

- server

entrypoint: /bin/sh

command: -c "curl http://server:3000/hello"

networks:

server:

driver: bridge

podman compose -f compose.yaml up -d

podman logs nodejs-client

Hello World!

# Experiment 3

*services:*

*server:*

*image: localhost/nodejs-server:latest*

*container\_name: nodejs-server*

*restart: always*

*ports:*

*- '8080:80'*

*networks:*

*- server*

client:

image: localhost/nodejs-client:latest

container\_name: nodejs-client

restart: always

depends on:

- server

networks:

- server

entrypoint: /bin/sh

command: -c "curl http://server:8080/hello"

networks:

server:

driver: bridge

Changed the ports of Client.js and server.js to 8080