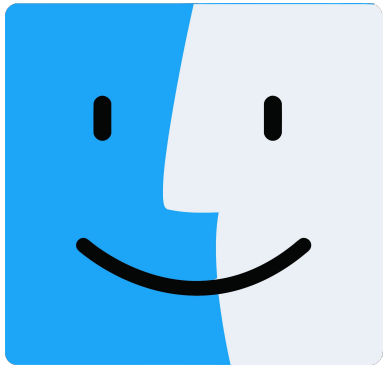


Podman on Intel MacOS

Podman and VM on Intel MacOS

DataSentics
an atos company

Michal Ježek
Cloud engineer

Machine Learning a Datova analyza

Aby data mela impact a pridanou hodnotu

Data-science workbench and containers

- Kubernetes on AWS EKS
- Jupyterhub in a pod
- Code-server in a pod

- Development of images on localhost
- Docker for enterprises is paid since new year

Podman

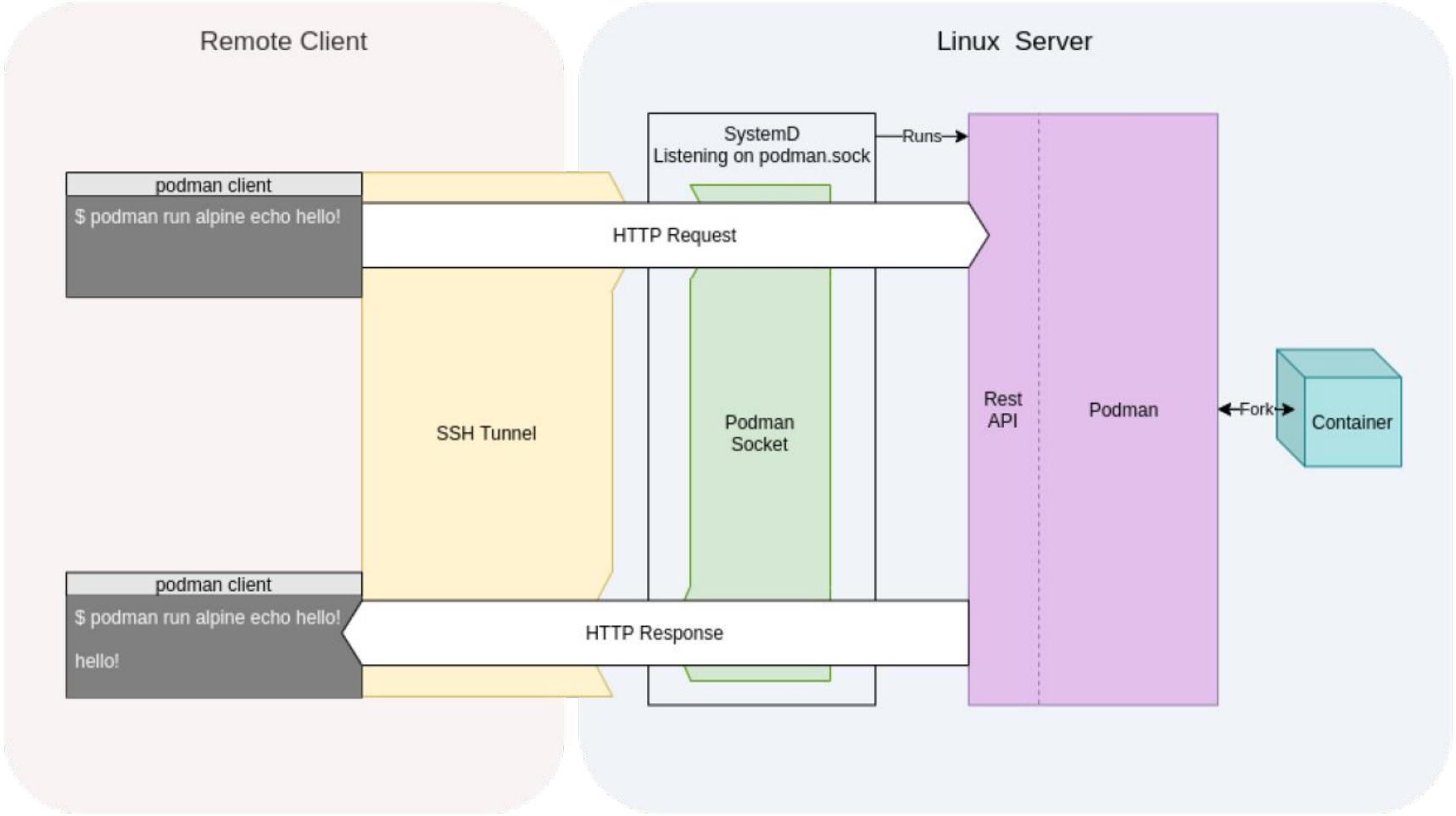
- Daemonless and rootless container manager
- Developed by Redhat
- Open Container Initiative compliant

- alias docker=podman

podman

Why VM ?

- Docker is build with Linux in mind
- Podman is the same
- There has to be Linux VM on MacOS running podman
- VM is running on QEMU in all cases (podman, lima, …)


Remote Client
Linux Server
SystemD
Listening on podman.sock
Runs
podman client
$ podman run alpine echo hello!
HTTP Request
SSH Tunnel
Podman
Socket
Rest
API
Podman
Fork
Container
podman client
$ podman run alpine echo hello!
hello!
HTTP Response

Lima

(Linux on Mac)

Virtual machine intended to bring nerdctl to MacOS, but works also for podman

- Automatic file sharing
- Automatic port forwarding
- Various guest Linux distributions
- Intel and ARM virtualization

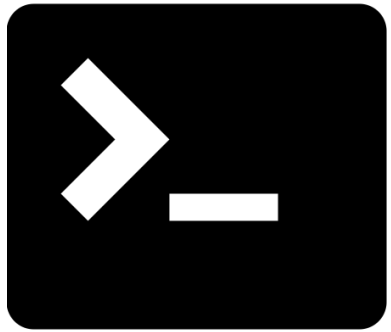
Why Lima

- Podman has it’s own virtualization (podman machine)
- Podman machine didn’t support the mounting of fs volumes
- Lima was able to mount the filesystem from MacOS
- … But Lima is using the sshfs => it’s pretty slow
- Applicable to version 0.11.0

Fedora in Lima VM

- Default Lima VM is ubuntu
- It is not creating the socket on MacOS with docker and podman host
- Fedora-podman.yml definition file is creating these sockets
  - https://github.com/containers/podman/issues/11533
- Sockets could be created also by ubuntu but more complex and lower podman version
- Sockets are used by VSCode and also needed for communication with other apps

Lima with Podman Usage



$ brew install podman

$ brew install lima

$ wget https://raw.githubusercontent.com/afbjorklund/fedora-lima/main/fedora-podman.yaml

$ limactl start ./fedora-podman.yaml

$ export CONTAINER_HOST=$(limactl list fedora-podman --format 'unix://{{.Dir}}/sock/podman.sock')

$ export DOCKER_HOST=$(limactl list fedora-podman --format 'unix://{{.Dir}}/sock/podman.sock')

—

$ podman run -it --rm -v "$PWD":/usr/src/myapp -w /usr/src/myapp docker.io/library/python:3 python hello_world.py

$ podman build -t alpine_moje .

$ podman run localhost/alpine_moje

Podman 4.0.0

- Since version of podman 4.0.0 it doesn’t cooperate with Lima
  - Lima : Error: unable to connect to Podman socket: server API version is too old. Client "4.0.0" server "3.4.4"
  - Workaround `brew pin podman` => it will not upgrade podman
- Podman has solved the issue with filesystem mounting
  - Possible to mount whole home directory with correct parameters
- Podman **build** is still not able to mount volumes

buildah

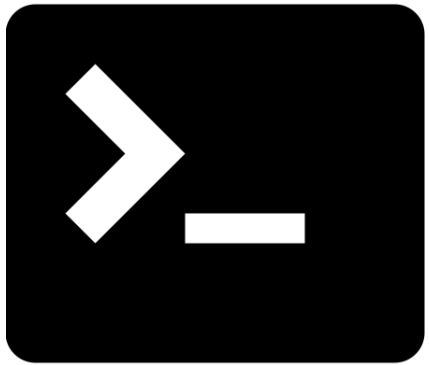
Podman machine init

```
$ podman machine init --cpus 1 --memory 4096 --disk-size 50 \
        --volume /Users --volume /Volumes <machine_name>

$ podman machine start <machine_name>

$ export DOCKER_HOST='unix:///Users/mjezek/.local/share/containers/podman/machine/podman-machine-default/podman.sock'

$ podman -c <machine_name> images
```



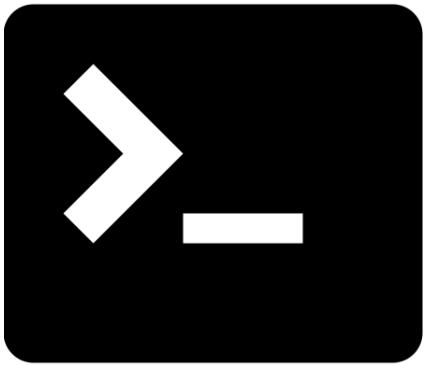
Podman latest version 4.1.0

Initialization of default podman machine

$ podman machine init

To change the configuration of initialized machine

$ podman machine set –cpus 4



Current practise

Use podman 4.0.0 or higher with podman machine (no Lima needed anymore)

Tested on Intel Mac only

Sharing the images in team

- ECR on AWS Cloud special CI/CD account
- Kubernetes takes the images from there
- Separation images with tags naming

In emergency situation :

- Podman load/save of images
- Transporting images with network storage to localhost
- due to long onboarding

Sharing images without registry

$ podman save -o nginx.tar docker.io/library/nginx

$ podman load -i nginx.tar

—

$ podman run docker-archive:nginx.tar echo hello

Error: unsupported transport docker-archive in "docker-archive:nginx.tar": only docker transport is supported

- Support of other docker formats for transporting the image

Remote container in VSCode on localhost

The env variable DOCKER_HOST has to be set

In settings for Remote-Containers :

- Docker path = podman
- Docker compose path = podman-compose

F1 -> Rebuild and Reopen in Container

-> Open Folder in Container

.devcontainer/devcontainer.json file has to be configured mainly with Dockerfile

.vscode/launch.json file for debugging and running the code

The podman issues

- podman **push/pull** do not support progress bars
  - not even with : $ podman machine ssh podman pull <image>
  - filtered by ssh : ssh -t should solve the issue
- podman **run** seems not to support docker-archive
  - not mentioned in documentation
- podman **build** do not support volume mounts
  - mentioned in documentation

- In podman machine Linux VM everything works