



podman desktop

Podman Desktop

Product Overview

Stevan Le Meur - Developer Tools PM



Local development environments
have become **impractical** and **lack**
consistency with production.



On the other hand, developers must deal with **higher complexity** and **additional overhead**.



podman desktop

Moving from Local to Prod

DEVELOPMENT TEAM



Inner Loop

LOCAL DEV ENV

- Base Images from Docker Hub
- Low / No Security
- Container Registry: Docker Hub
- Docker Compose
- Nonbinding with Managed Services
- Minikube

Heavy Usage of:

WALL OF DISCREPANCIES

KUBERNETES / OPENSOURCE

- UBI base Images
- Quay.IO
- Rootless
- Kubernetes YAML
- Managed Services

OPS TEAM



Reproducing "Prod workloads" environment in Local



The consequence:

A **huge adoption barrier of Kubernetes/ technologies**, complicated to reconcile developer environments with targeted production environment constraints.

Developers are using tools and technologies that keep them away from the target production environment.



Podman Desktop **simplify** workflows and experiences working with containers when targeting production on **Kubernetes / OpenShift** by:

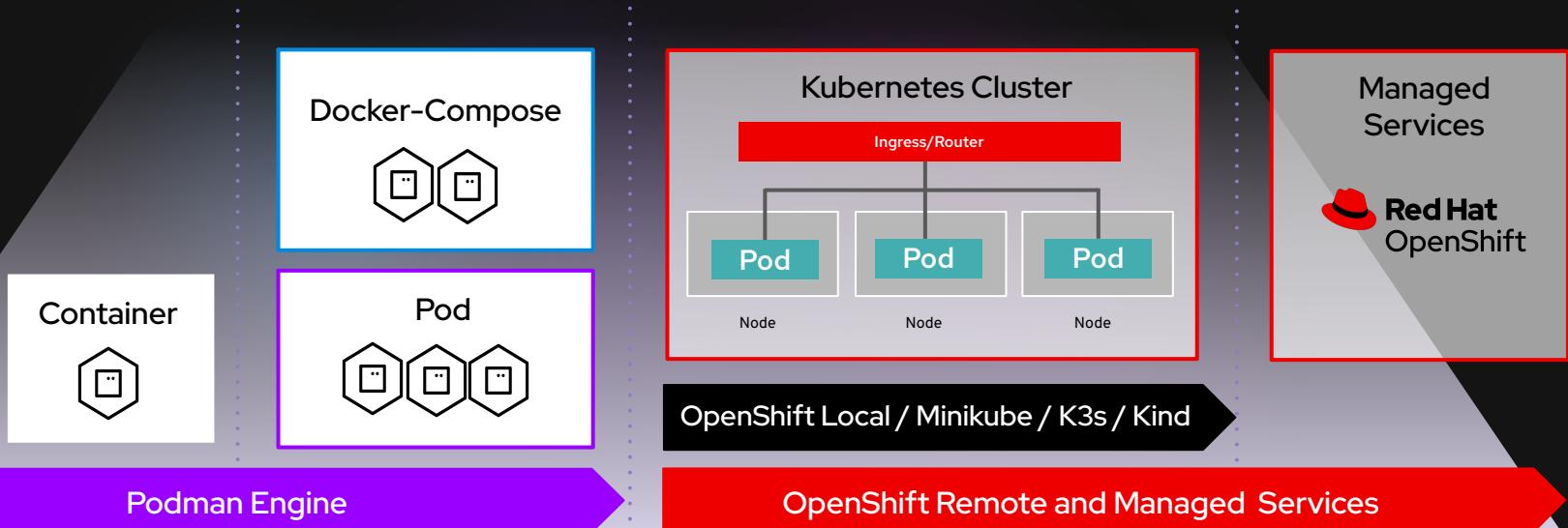
- Bringing Kubernetes closer to the developer
- Minimizing the discrepancies between the desktop and target Kubernetes environments.



podman desktop

Simplistic onboarding.

From applications to containers, to pods, to platforms to **OpenShift**.



podmandesktop



podman desktop

Introducing Podman Desktop

Containers and Kubernetes for Application Developers

Podman and Kubernetes/OpenShift Local

- Install and run anywhere: Windows, Mac and Linux
- Keep it up-to-date

Containers and Pods

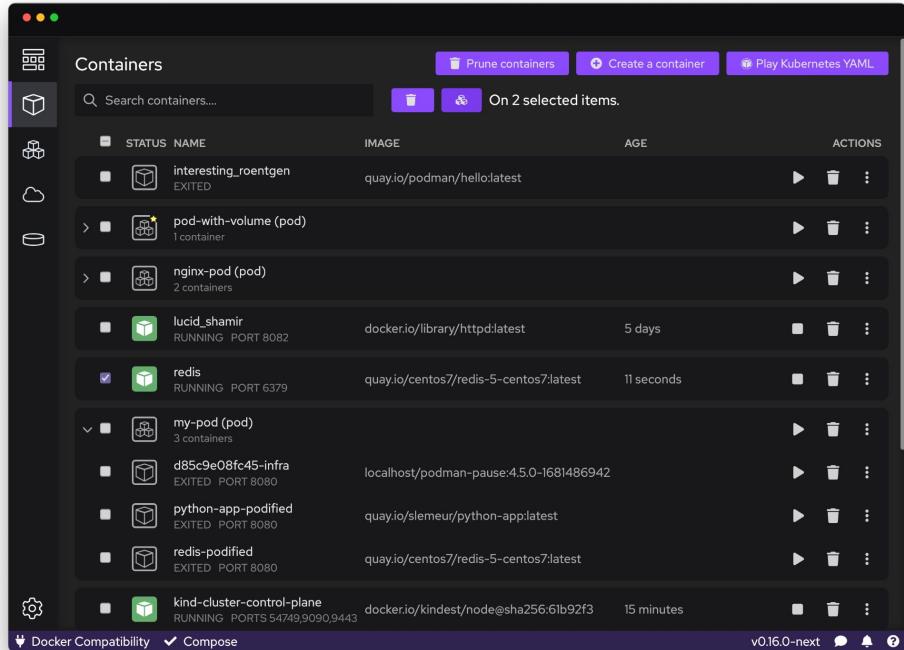
- Build, run, manage and debug Containers and Pods
- Run Pods with or without Kubernetes
- Manage multiple container Engines
- Compatibility with Docker Compose

Enterprise Readiness

- VPN and Proxies configuration
- Image registry management
- AirGapped Installation

Bridge between local and remote

- Connect and deploy to remote OpenShift clusters
- Enable remote managed services locally





podman desktop

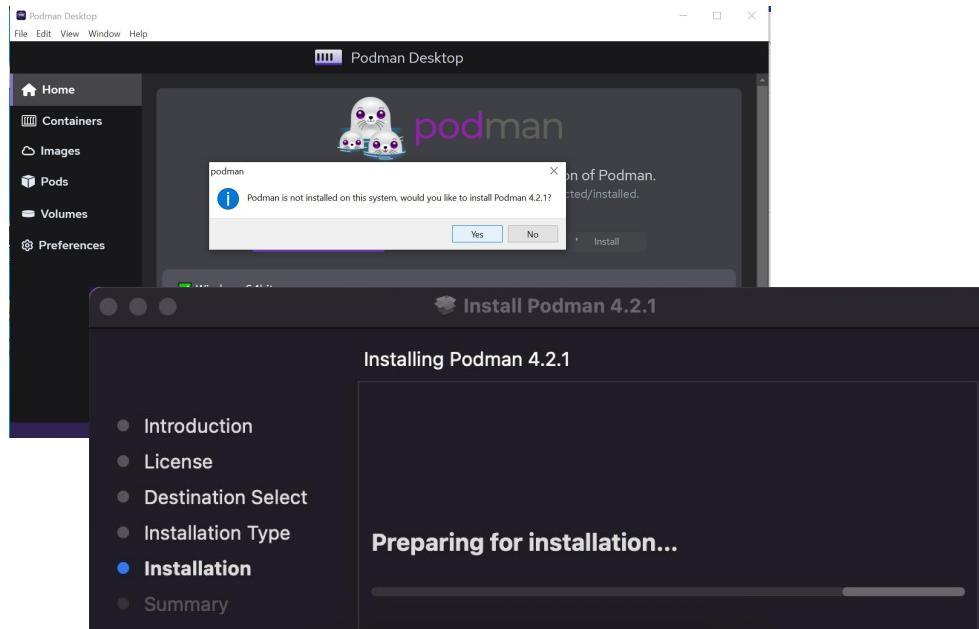
Demo



podman desktop

Cross-platform Installation of Podman Container Engine

- Run anywhere:
 - **Windows** (*EXE, Chocolatey, Winget, Scoop*)
 - **Mac** (*DMG or Brew*)
 - **Linux** (*Flathub, Flatpack, Zip*)
- Install and keep-up-to date Podman Engine
- Configure and initialize Podman

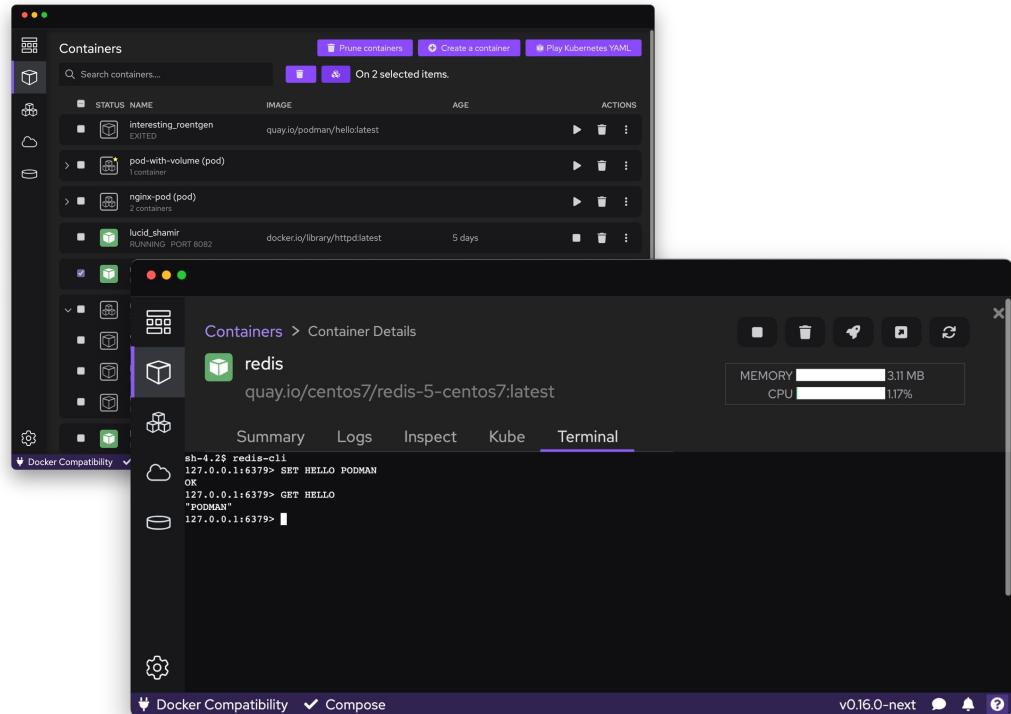




podman desktop

Management of Containers

- Build images from Dockerfile or Containerfile
- Run, test, debug containers
- Run Compose files
- Built-in Terminal to SSH into containers
- Inspect Containers Logs
- Manage Volumes





podman desktop

Support for OCI Registries

- Configure multiple OCI registries
- Authenticate to registries
- Pull, tag and push images to your registries

The screenshot shows the Podman Desktop interface. On the left is a sidebar with icons for Settings, Resources, Proxy, **Registries** (which is selected), Authentication, Extensions, Desktop Extensions, and Preferences. The main area has two panes: the left pane displays a list of registries with their locations, usernames, and password fields; the right pane shows a list of images with columns for AGE, SIZE, and ACTIONS. A modal dialog is open in the center, titled 'Registries', showing configuration details for Red Hat Quay, Docker Hub, GitHub, and Google Container Registry, along with fields for URL, Username, Password, and a 'Configure' button.

Registry Location	Username	Password
Red Hat Quay	slemeur
Docker Hub	stevanim
GitHub		Configure
Google Container Registry		Configure

Images

AGE	SIZE	ACTIONS
5 days	927.75 MB	Push Image
5 days	356.48 MB	Show History
6 days	959.11 MB	Push image to Kind cluster
6 days	800.14 MB	
2 weeks	655.7 MB	
2 weeks	149.58 MB	
2 weeks	146.55 MB	
2 weeks	120.7 MB	v0.16.0-next



podman desktop

Pods

- Create and start Pods with Podman
- Select containers to run as a Pod
- Play Kubernetes YAML locally without Kubernetes
- Generate Kubernetes YAML from Pods

The screenshot shows the Podman Desktop application interface. The main window title is "Pods". The left sidebar has icons for "Containers", "Pods", "Cloud", and "Docker". The main area displays a list of pods:

STATUS	NAME	AGE
Running	simple-pod 43c81ac5 1 container	5 days
Running	pod-with-volume 62ecfc06 1 container	5 days
Running	nginx-pod acf41e0 2 containers	5 days
Running	my-pod d85c9e08 3 containers	5 days

A context menu is open over the "simple-pod" entry, showing options like "Copy containers to a pod", "Prune pods", and "Play Kubernetes YAML". The "Copy containers to a pod" submenu is expanded, showing:

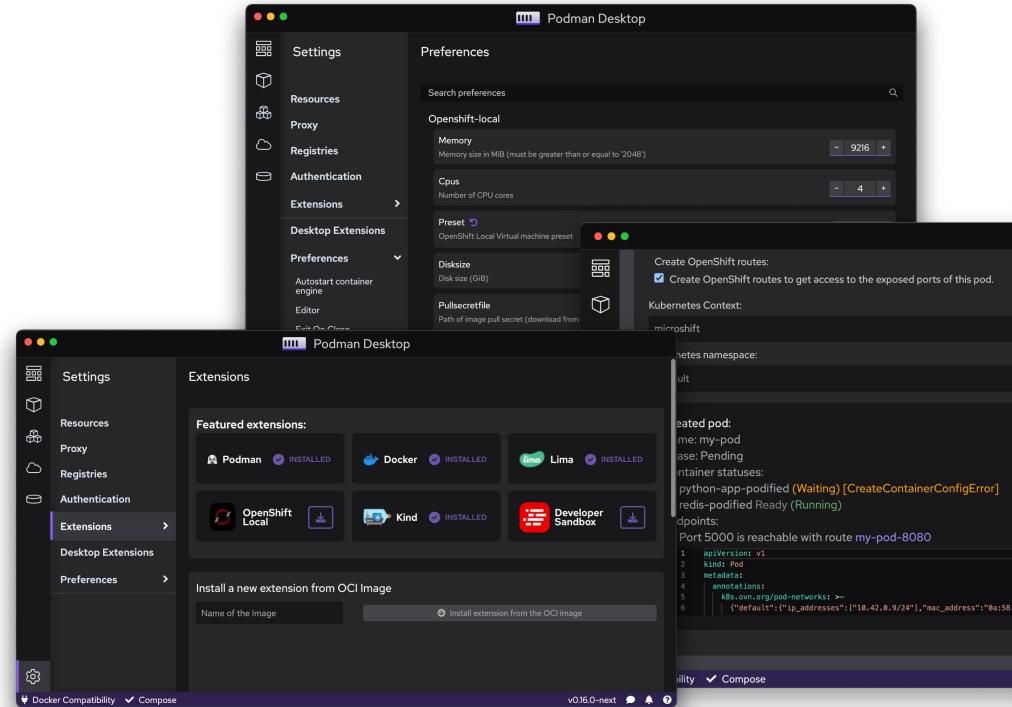
- Name of the pod: my-pod
- Containers to replicate to the pod:
 - 1. redis (2d6e04a)
 - 2. python-app (487f377)
- All selected ports will be exposed:
 - Port 6379 redis
 - Port 8080 python-app

At the bottom of the main window, there are buttons for "Docker Compatibility" and "Compose". The status bar at the bottom right shows "v0.16.0-next" and some notification icons.



OpenShift and Dev Sandbox Integration

- OpenShift Local extension
 - **Light and optimized** - Powered by Microshift (experimental)
 - For Developers
 - Minimal services set
 - Fast and lightweight
 - **Single-node OpenShift** - Powered by OpenShift Container Platform
 - Full services set
 - Complete and more resource-intensive
- Support for Dev Sandbox

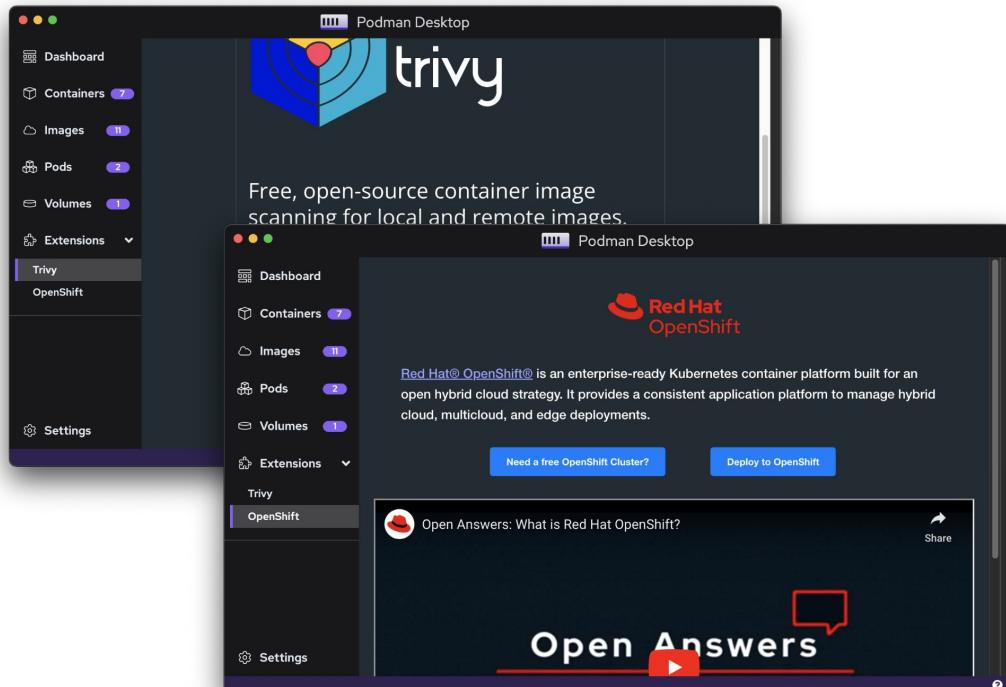




podman desktop

- Support for multiple Container Engines
 - Podman
 - Docker
 - Lima
 - CRC / OpenShift Local
 - Future local light distributions of Kubernetes
- Support for Docker Desktop Extensions

Open: By default





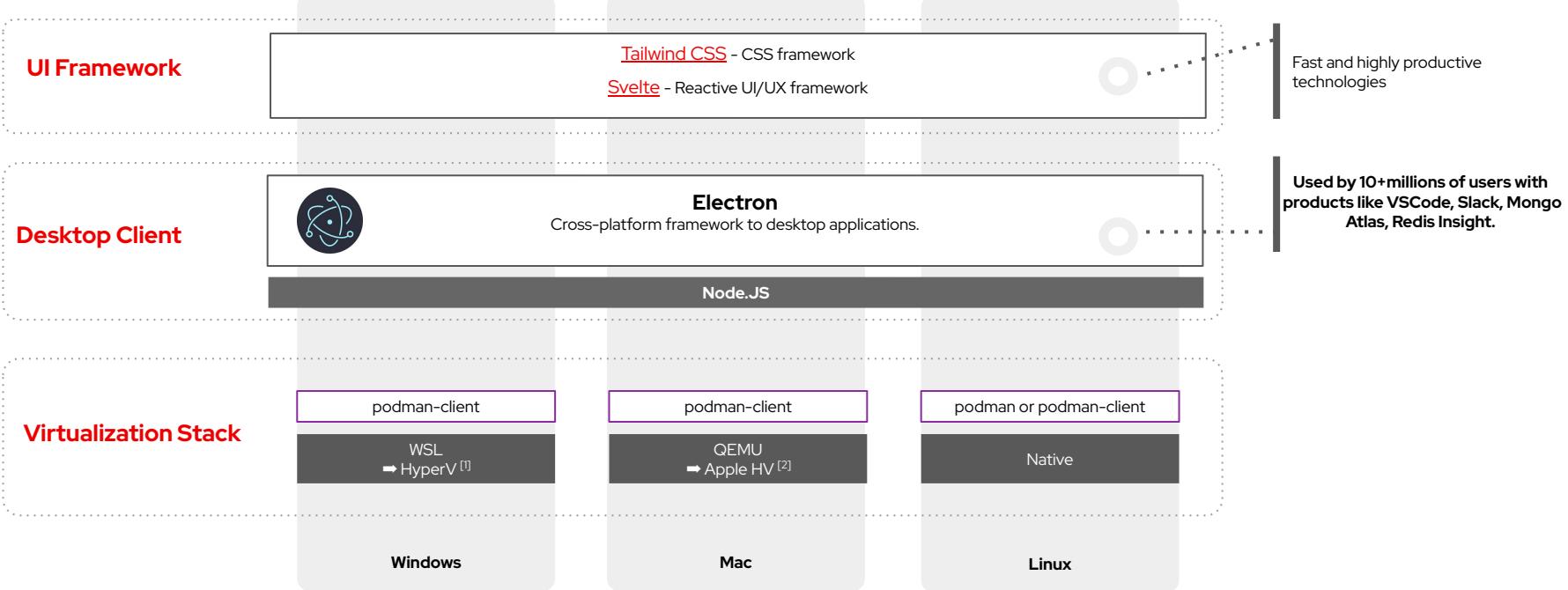
podman desktop

Free, Open and Extensible

-

By Default

Building with proven foundations and leveraging our experience

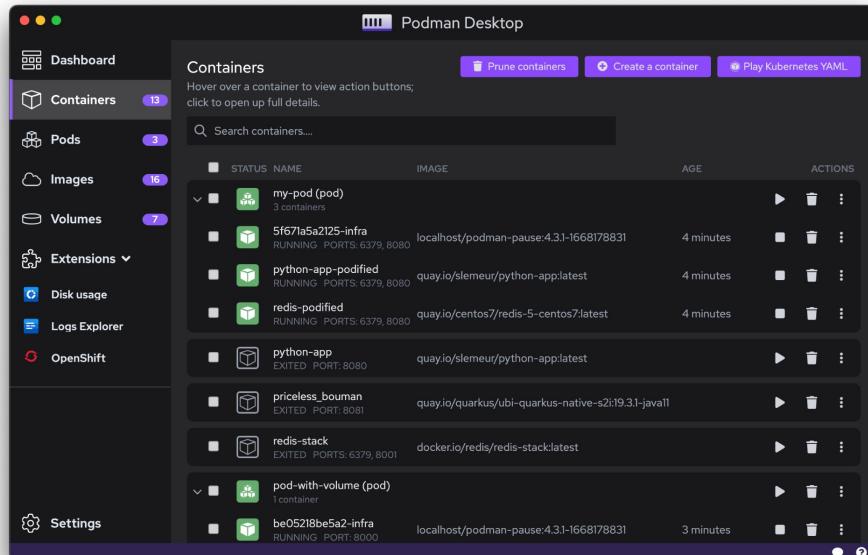


[¹] HyperV under active development, targeted for podman 4.5
[²] Apple Hypervisor support in early (but active) planning

Extensibility

From extension points to other container/K8s technologies

Supports Docker Desktop extensions



Current extensions:



Podman



Kind

Lima



Docker



OpenShift Local

But Podman Desktop extensions can also do much more:

- Container engine providers
- Kubernetes providers
- Add actions
- Add menus
- Add configuration
- Add default registries
- Add to status bar
- Add to system tray
- ...

Default
Registries



podman desktop

Support Model



Envisioned Support Model

- Podman Desktop is a community project and not an official Red Hat product
- Podman Desktop will get “[Enhanced Community Support](#)” for any Red Hat customers (RHEL, including ARO, ROSA, OSD, OpenShift managed by IBM)
 - Non-production support
 - More than community support
 - No/limited SLA - best effort ('commercially reasonable support', in legal-speak), business hours
 - Only entitled customers can use this level to open cases - namely OCP or OSD customers. Feedback from Px working session:
It would be best if the Portal interface could differentiate between 'Fully Supported Products' and 'Enhanced Community Support Offerings'
 - Will take salesforce tickets
 - Will transfer to appropriate product engineering team and open JIRA/BZs
- Podman has supported option on RHEL



podman desktop

Preparing Podman Desktop to be a Docker Desktop replacement for Quarkus

Install podman

```
$ brew install podman  
$ brew install podman-desktop  
$ podman machine init --cpus 2 -m 4096 --rootful  
$ podman machine start
```

Prepare TestContainers for Quarkus

```
$ echo "ryuk.container.privileged=true" >> ~/.testcontainers.properties
```

Now you can use podman for Quarkus Dev Services



Roadmap

Podman Desktop

Short Term

(3-6 months)

Dashboard:

- Onboarding Experience
- Better Settings/Configuration Management
- Networks

Kubernetes Integration:

- Kind Support
- Enhanced transition from Containers to Pods
- Enhanced transition from Pods to Kubernetes

Container Tooling:

- Compose support

Red Hat Integration:

- Option for installing OpenShift Local
- Integration with Red Hat Developer Sandbox
- Image OpenShift Readiness Checks

Podman Desktop

Mid Term

(6-9 months)

Kubernetes Integration:

- From Compose to Kube
- Bridge with Kubernetes workloads

System Tray:

- Display Resource Utilization

Dashboard:

- Kubernetes workload explorer
- Dashboard with Statistics

OpenShift Local:

- Microshift for Developers

Red Hat Integration:

- Red Hat Container Catalog

Podman Desktop

Long Term

(9 months+)

Continuation from the previous items

+

Tell us your problems and what you need to be solved!

Podman Desktop



podman desktop

Learn More



podman desktop

Start Using

Podman-desktop.io

Podman.io

Contribute, report issues:

github.com/containers/podman-desktop