



podman desktop

# Podman Desktop

**Product Overview**

Wanja Pernath



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:

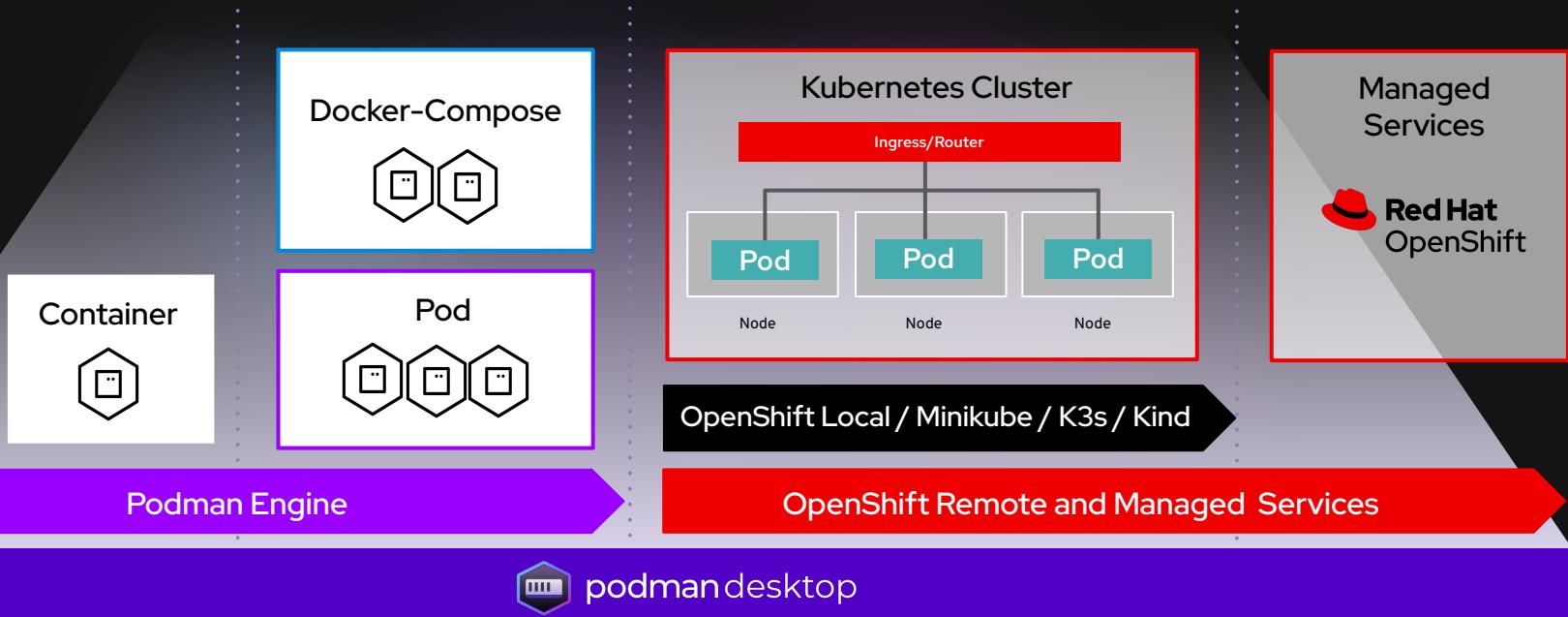
- Bringing Kubernetes closer to the developer
- Minimizing the discrepancies between the desktop and target Kubernetes environments.



podman desktop

# Simplistic onboarding.

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





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

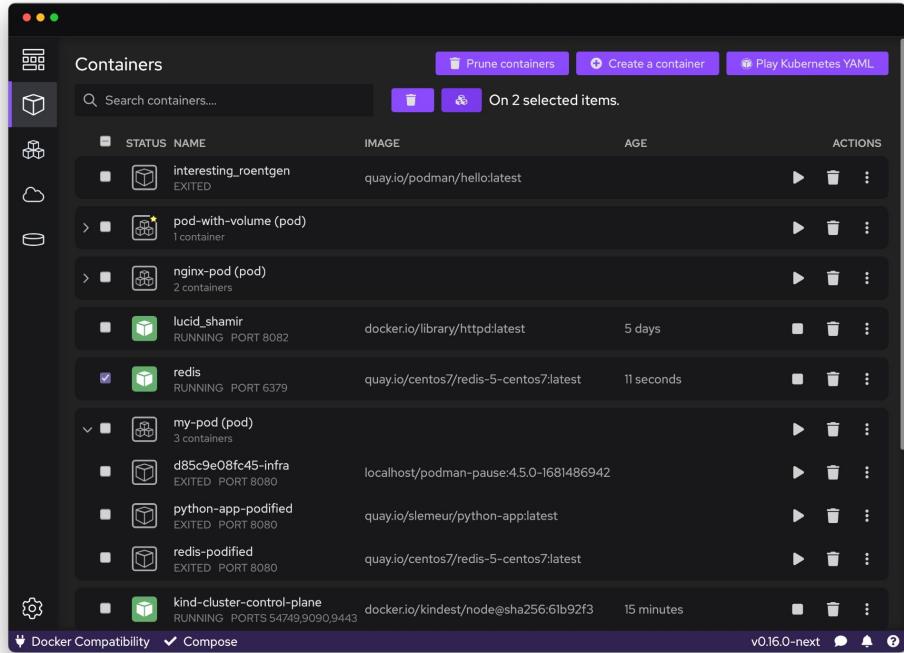
- Build, run, manage and debug Containers and Pods
- Run Pods with or without Kubernetes
- Manage multiple container Engines
- Compatibility with Docker Compose

### 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

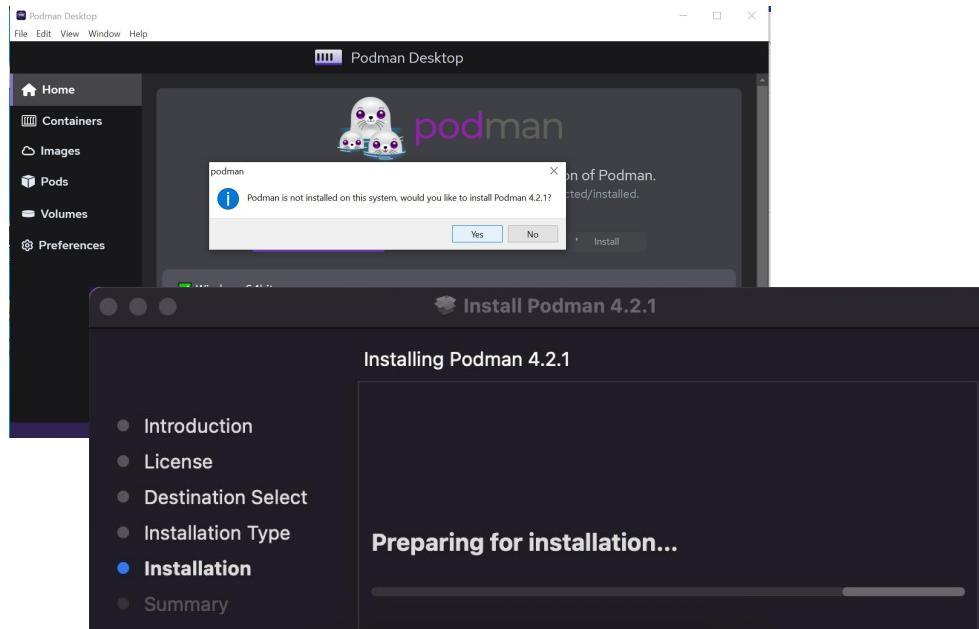
Demo



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

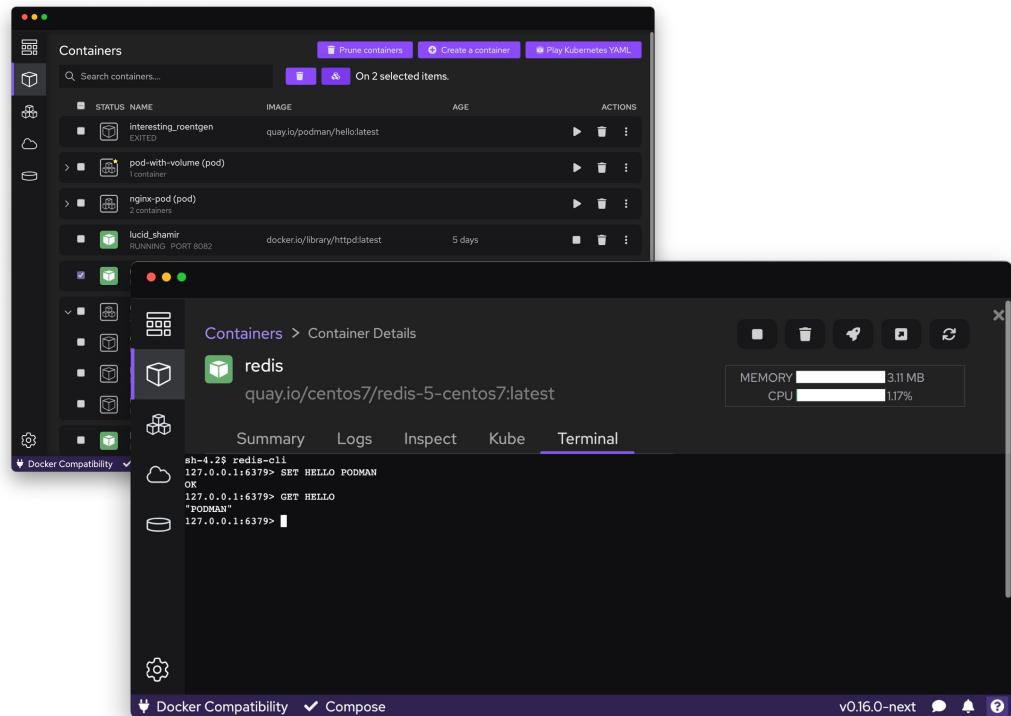




# podman desktop

## 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. Below the sidebar is a status bar with Docker Compatibility and Compose icons, and the version v0.16.0-next.

The main area has two tabs: "Images" (selected) and "Registries". The "Images" tab shows a list of images with columns for AGE, SIZE, and ACTIONS. One item is highlighted: "quay.io/slemeur/python-app alec26a0302f latest" (5 days, 927.75 MB). Action buttons include "Push Image", "Show History", and "Push image to Kind cluster".

The "Registries" tab shows a list of configured registries with columns for Registry Location, Username, and Password. It includes entries for Red Hat Quay (slemeur), Docker Hub (stevanim), GitHub, and Google Container Registry. Buttons for "Configure" and "Add registry" are present.



# 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 is titled "Pods" and displays a list of existing pods: "simple-pod" (1 container), "pod-with-volume" (1 container), "nginx-pod" (2 containers), and "my-pod" (3 containers). Each pod entry includes its status, name, and age. A search bar at the top allows users to find specific pods. On the right side of the main window, there are several icons for actions like "Prune pods" and "Play Kubernetes YAML". A modal dialog is open, titled "Copy containers to a pod", which is used to create a new pod named "my-pod". Inside the modal, a list of containers to replicate is shown: "redis" and "python-app". Below this, a section for exposing ports is visible, with checkboxes for "Port 6379" and "Port 8080", both of which are checked. At the bottom right of the modal, there are "Close" and "Create Pod" buttons. The bottom of the screen shows the application's footer with icons for Docker Compatibility and Compose, and the text "v0.16.0-next".



# podman desktop

## OpenShift and Dev Sandbox Integration

### ➤ OpenShift Local extension

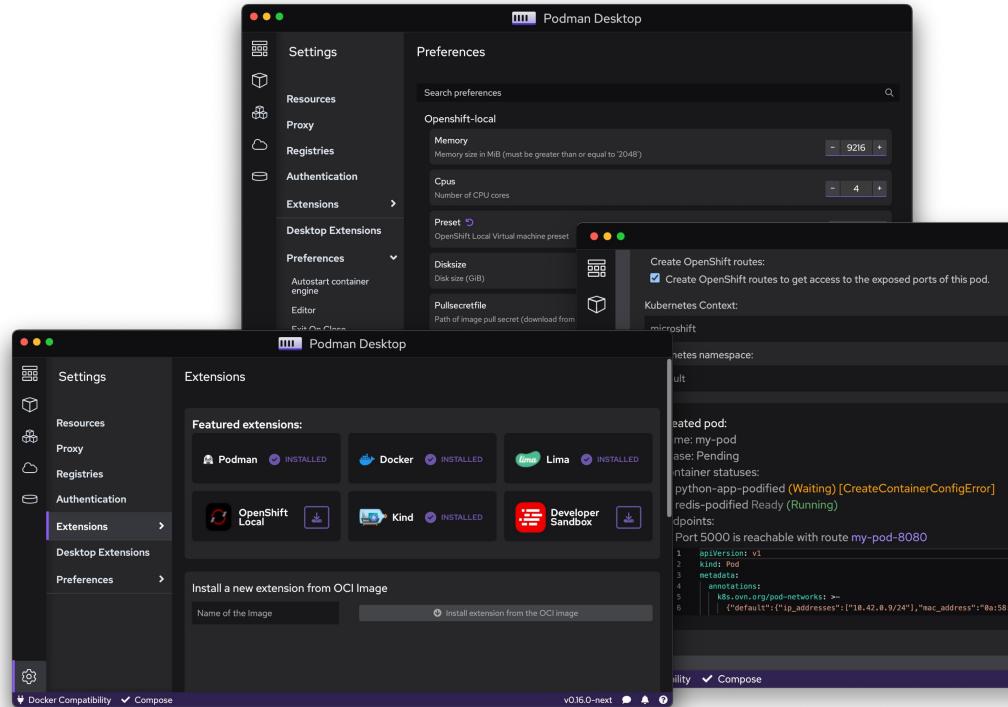
- **Light and optimized** - Powered by Microshift  
(experimental)
  - For Developers
  - Minimal services set
  - Fast and lightweight

- **Single-node OpenShift** - Powered by

OpenShift Container Platform

- Full services set
- Complete and more resource-intensive

### ➤ Support for Dev Sandbox

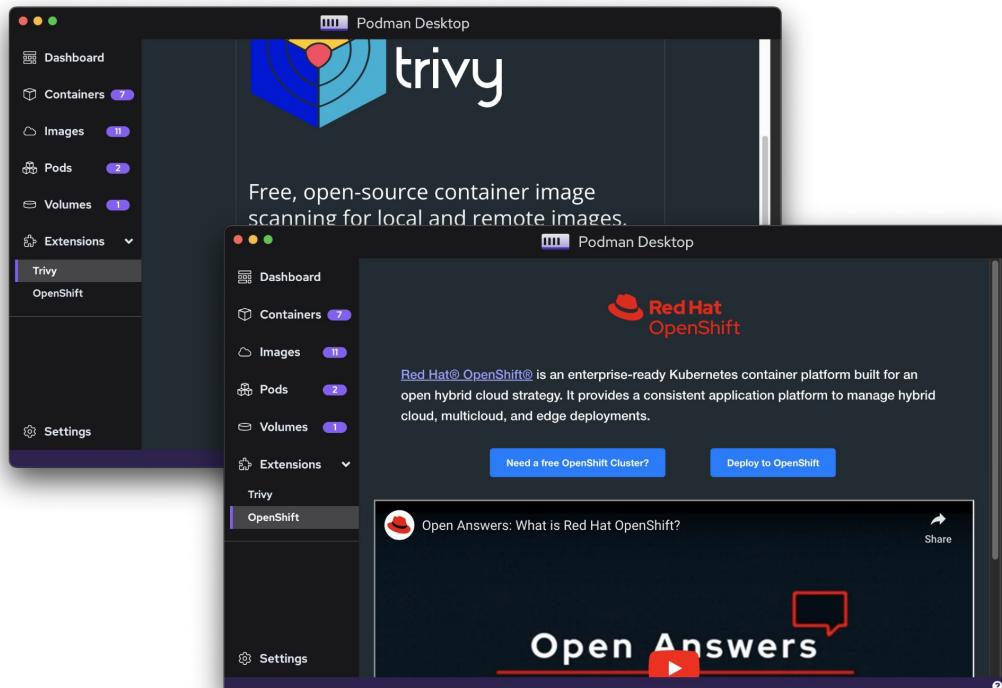




# podman desktop

- 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

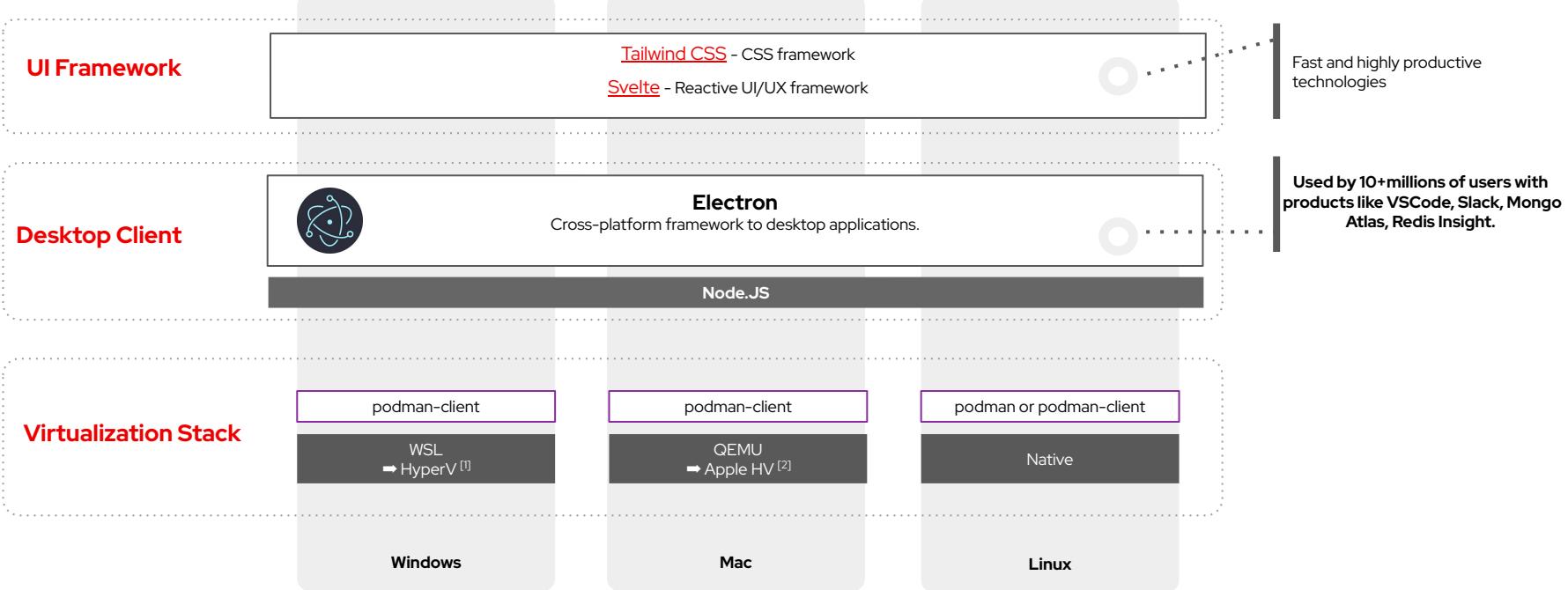




podman desktop

Free, Open and Extensible  
-  
**By Default**

# Building with proven foundations and leveraging our experience

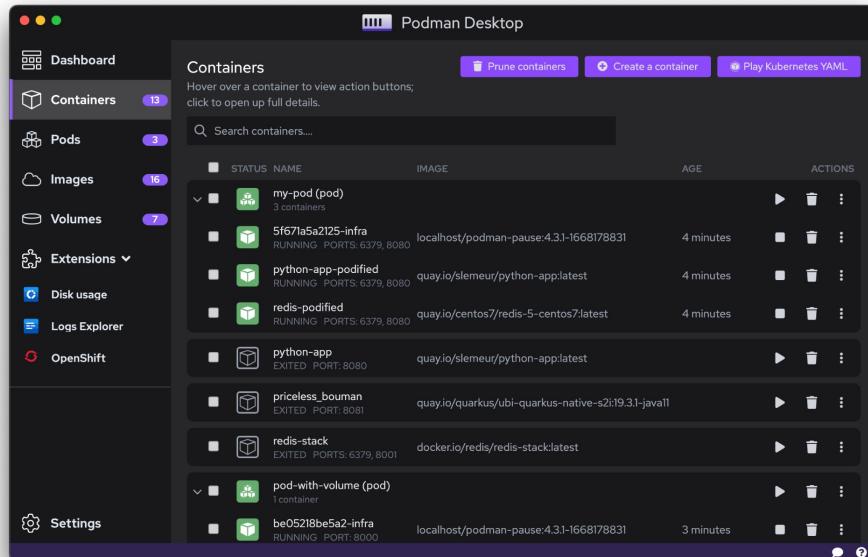


<sup>[1]</sup> HyperV under active development, targeted for podman 4.5  
<sup>[2]</sup> Apple Hypervisor support in early (but active) planning

# 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



podman desktop

# 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](https://Podman-desktop.io)

[Podman.io](https://Podman.io)

## Contribute, report issues:

[github.com/containers/podman-desktop](https://github.com/containers/podman-desktop)