Installing Rancher Desktop

macOS

Prerequisites

Rancher Desktop requires the following on macOS:

- macOS Catalina 10.15 or higher.
- Apple Silicon (M1) or Intel CPU with VT-x.
- Persistent internet connection.

It is also recommended to have:

- 8 GB of memory
- 4 CPU

Additional resources may be required depending on the workloads you plan to run.

Installation

1. Go to the releases page on GitHub.
2. Find the version of Rancher Desktop you want to download.
3. Expand the **Assets** section and download `Rancher.Desktop-X.Y.Z.dmg`, where `X.Y.Z` is the version of Rancher Desktop.
4. Navigate to the directory where you downloaded the installer. This will usually be the `Downloads` folder.
5. Double-click the DMG file.
6. In the Finder window that opens, drag the Rancher Desktop icon to the Applications folder.
7. Navigate to the `Applications` folder and double-click the Rancher Desktop to launch it.

Windows

Prerequisites

Rancher Desktop requires Windows Subsystem for Linux on Windows; this will automatically be installed as part of the Rancher Desktop setup. Manually downloading a distribution is not necessary.

It is also recommended to have:

- 8 GB of memory
- 4 CPU

Additional resources may be required depending on the workloads you plan to run.

Note: You can use Rancher Desktop as a Non-Admin user on a Windows machine. However, an Admin's intervention is required during the installation process for the below components.

- **WSL2 -** You need Admin privileges to install WSL2, which is an essential component of Rancher Desktop.
- **Rancher Desktop Privileged Service -** You need Admin privileges to install the Rancher Desktop privileged service, which is required to expose applications/services, running inside containers, on all interfaces on the host machine. However, you can skip the installation of the Rancher Desktop Privileged Service with the limitation that you will not be able to expose applications/services on any interface except `127.0.0.1`.

Installation

1. Go to the releases page on GitHub.
2. Find the version of Rancher Desktop you want to download.
3. Expand the **Assets** section and download the Windows installer. It will be called `Rancher.Desktop.Setup.X.Y.Z.msi`, where `X.Y.Z` is the version of Rancher Desktop.
4. Navigate to the directory where you downloaded the installer to and run the installer. This will usually be the `Downloads` folder.
5. Review the License Agreement and click **I Agree** to proceed with the installation.
6. If prompted, choose between installing for everyone on the machine or installing just for the current user. Installing for everyone is preferred in order to install the Rancher Desktop Privileged Service, as noted above.
7. Follow the prompts to confirm installation.
8. When the installation completes, click **Finish** to close the installation wizard.

Test Rancher Desktop

Hello World Example

Get started with Rancher Desktop by pushing an app to a local Kubernetes cluster.

Rancher Desktop works with two container engines, containerd and Moby, the open-sourced components of the Docker ecosystem. For `nerdctl`, use the **containerd** runtime. For `docker`, use the **dockerd(moby)** runtime.

Example#1 - Build Image & Run Container

Create a folder

```
mkdir hello-world
cd hello-world
```

Create a blank Dockerfile

On Windows, Create a blank file named `Dockerfile`

On macOS or Linux, You can use the below command to create a blank `Dockerfile`

```
vi Dockerfile
```

Populate the Dockerfile with the command below

```
FROM alpine
CMD ["echo", "Hello World!!"]
```

Build and run the image for verification purposes

```
docker build --tag helloworld:v1.0 .
docker images | grep helloworld
docker run --rm helloworld:v1.0
# Remove the image
docker rmi helloworld:v1.0
```

Example#2 - Build Image & Deploy Container to Kubernetes

Make sure you switch the Container Runtime setting in the Kubernetes Settings panel to `dockerd` or `containerd` as needed.

Create a folder and add a sample index.html file as follows

```
mkdir nginx
cd nginx
echo "<h1>Hello World from NGINX!!</h1>" > index.html
```

Create a blank Dockerfile

On Windows, Create a blank file named `Dockerfile`

On macOS or Linux, You can use the below command to create a blank `Dockerfile`

```
vi Dockerfile
```

Populate the Dockerfile with the command below

```
FROM nginx:alpine
COPY . /usr/share/nginx/html
```

Build image from code locally

```
docker build --tag nginx-helloworld:latest .
docker images | grep nginx-helloworld
```

Deploy to Kubernetes

Run the below command to create and run a pod using the image built in the previous step.

⚠️ **Note:** Please note that you need to pass the flag `--image-pull-policy=Never` to use a local image with `:latest` tag, as `:latest` tag will always try to pull the images from a remote repository.

```
kubectl run hello-world --image=nginx-helloworld:latest --image-pull-policy=Never --
port=80
kubectl port-forward pods/hello-world 8080:80
```

Point your web browser to `localhost:8080`, and you will see the message `Hello World from NGINX!!`. If you prefer to stay on the command line, use `curl localhost:8080`.

Delete the pod and the image

```
kubectl delete pod hello-world
# Remove the image
docker rmi nginx-helloworld:latest
```