

VIS 2019

Running TTK with Docker

Christoph Garth

TU Kaiserslautern



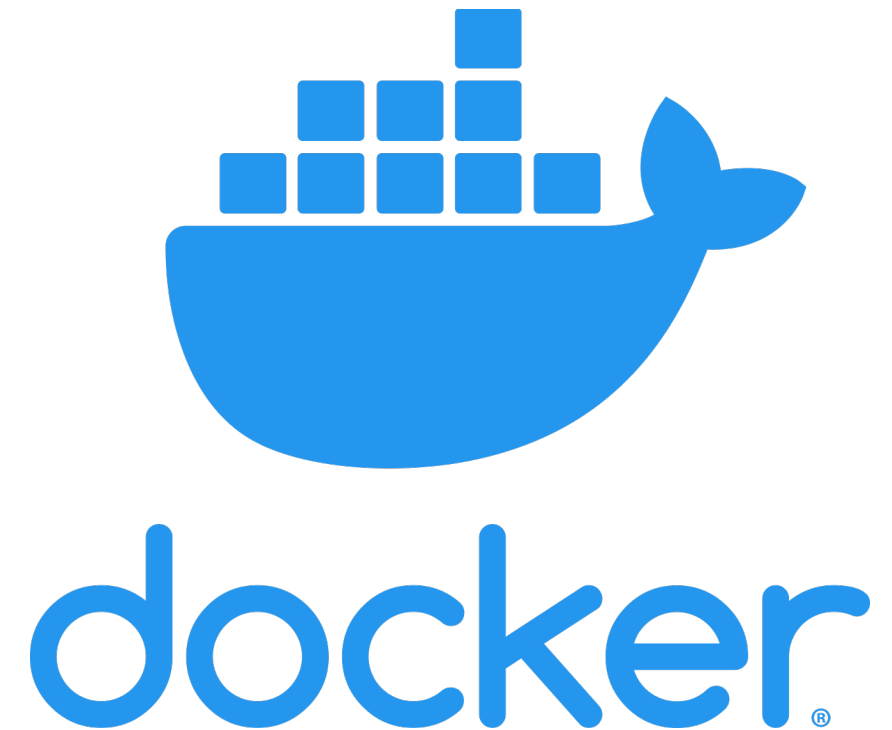


Docker in 3 minutes

What is Docker?

Docker is a tool designed to make it easier to create, deploy, and run applications by using containers. Containers allow a developer to package up an application with all of the parts it needs, such as libraries and other dependencies, and ship it all out as one package.

– from opensource.com



Why use Docker?

- Dependency problems are a bane encountered in scientific workflows
 - conflicting dependencies for installed binaries
 - frequent recompilation needed
 - platform-specific problems
- Docker containers allow packaging of TTK with all dependencies
 - Use TTK from container instead of directly from native install.
 - No installation, no dependency problems, no recompiles.

Docker Essential Terminology

- A **Docker container** is a lightweight, encapsulated environment
 - almost completely isolated from the hosting operating system; similar to a “thin” virtual machine (no hardware emulation)
 - executed on the **host system** with OS support (Linux) or in a Linux VM (Windows, macOS)
- A **Docker image** represents a snapshot of a Docker container
 - Running a container initializes it from an image.
- The **Docker engine** transparently takes care of executing containers.

Getting Docker

Docker is not open source, but the [Community Edition](#) is free to use.

- Requires superuser / administrator privileges on the host system.

Installation instructions for Docker Community Edition:

Linux

[Ubuntu](#), [Debian](#), [CentOS](#),
[Fedora](#), [other Linux](#)

QoL improvement:
run containers without sudo

Windows

Windows 10:
[Docker Desktop for Windows](#)

Older Windows:
[Docker Toolbox for Windows](#)

macOS

10.12 and later:
[Docker Desktop for Mac](#)

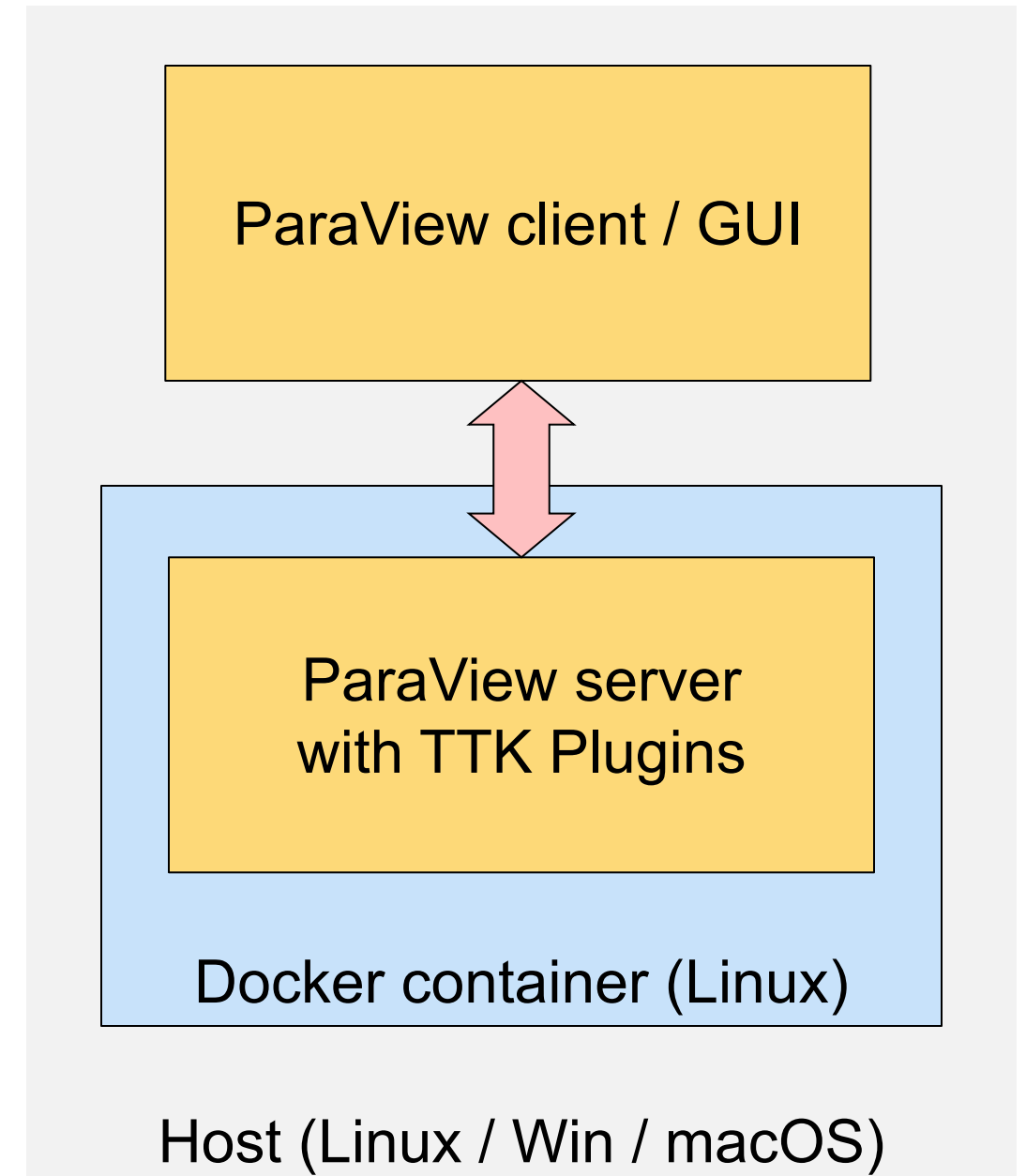
Older:
[Docker Toolbox for Mac](#)



TTK+ParaView Docker Containers

Setup

- TTK+ParaView Docker containers utilize ParaView's built-in client/server mode
 - Server (including TTK plugins) runs in container.
 - Default ParaView client (GUI) runs in host OS.
 - Communication via host↔container networking.
 - No compilation needed at all.
- Caveats
 - Client / container versions must match exactly
 - Only software rendering and OSPRay supported; no hardware acceleration possible



Running the TTK+Paraview Docker Image

Assume ParaView client / GUI 5.6.1 installed; want TTK version 0.9.8

Enter in terminal:

```
docker run -it --rm -p 11111:11111 -v ${HOME}:${HOME} -u ${UID} topologytoolkit/ttk:5.6.1-0.9.8
```

Remove container after exit

Same user in container as on host.

Image to run.

Run interactively (allow
Ctrl-C).

Allow container to receive network
connections on port 11111

Map user home directory to same
path in container.

Running the TTK+Paraview Docker Image

Alternative: convenience shell script (should work most of the time).

Enter in terminal:

```
cd <path to TTK source>  
scripts/docker/runParaViewTTKdocker.sh
```

Attempts to auto-detect ParaView installation and runs matching container.
(Can also supply ParaView binary as argument.)

Running the TTK+Paraview Docker Image

Alternative: Execute Python script in container.

Enter in terminal:

```
cd <path to TTK source>  
scripts/docker/runTTKPythonDocker.sh <script>
```

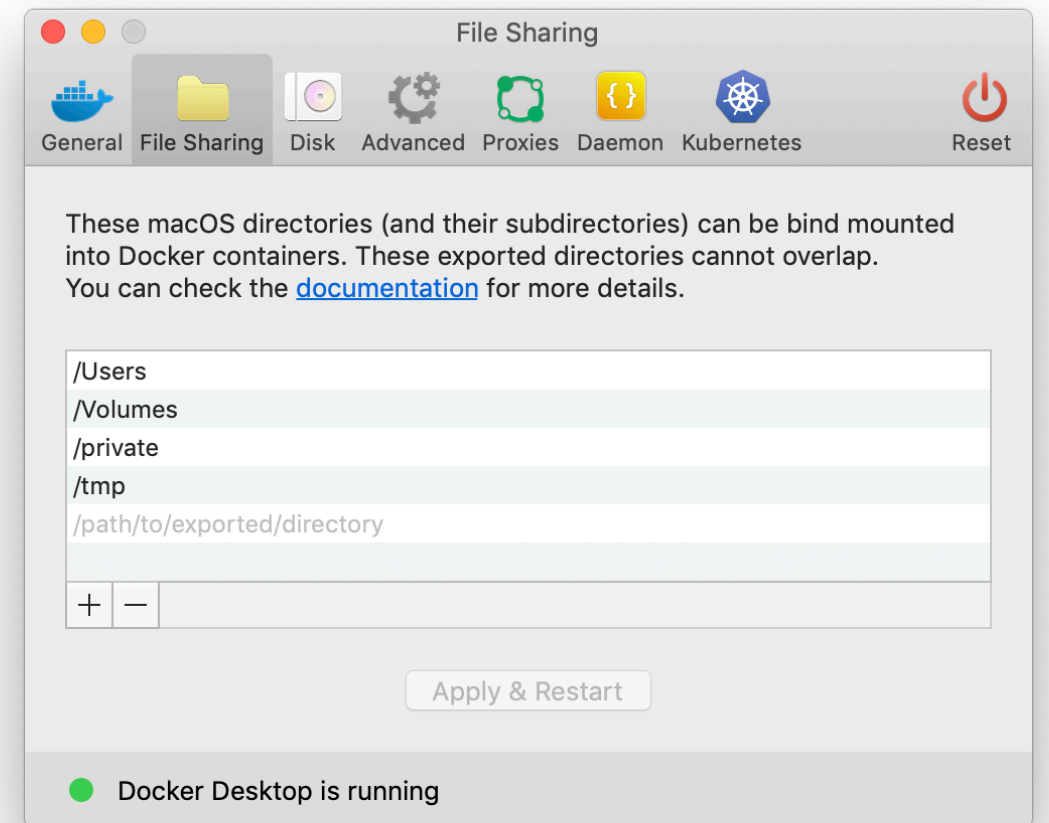
Attempts to auto-detect ParaView installation and runs matching container.
(Can also supply ParaView binary as argument.)



Live Demo

Notes

- File paths (data or other) must be made available to container.
 - Host filesystem is not visible to container by default.
 - Must explicitly pass "-v" flags for needed directories.
 - Convenience script will automatically do this for home directory.
- Windows + macOS: container inside virtual machine
 - Docker Desktop Preferences:
set up paths in "Shared Files" or "File Sharing"

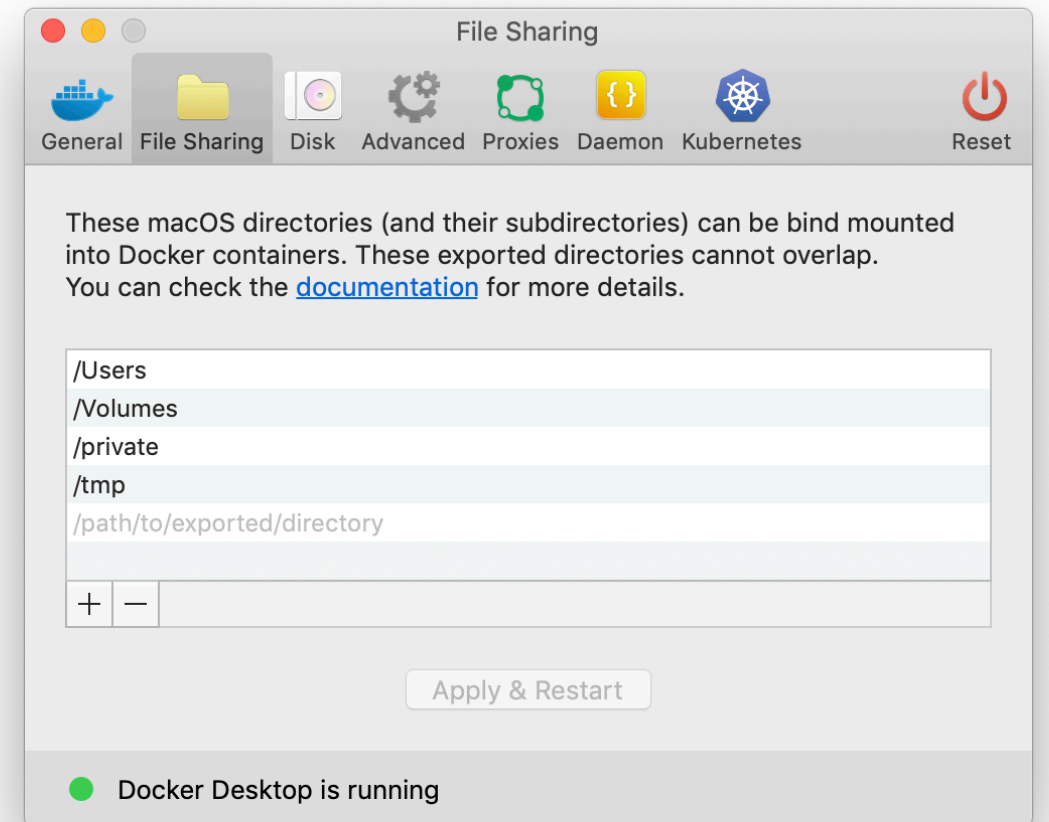


Notes II

- Container can also run on other host
 - Just like ParaView server without container
 - Need superuser privileges however
- Build your own container
 - All scripts in <TTK>/scripts/docker
 - E.g.

```
cd <TTK>/scripts/docker
```

```
docker build -t topologytoolkit/ttk:5.6.1-0.9.8 \  
--build-arg ttk=0.9.8 .
```



Conclusion

Main message:

TTK+Paraview Docker containers can be an easy way to use TTK without manual installation of TTK and dependencies.

Look at `<TTK>/scripts/docker` for container build scripts, etc.

Improvements and contributions welcome!