Docker Images for Software in SPP Computeralgebra

Sebastian Gutsche

TU Kaiserslautern

October 1, 2015

What is Docker?

Q: What is Docker?

What is Docker?

Q: What is Docker?

A: An enviroment for virtual machines.

What is Docker?

Q: What is Docker?

A: An enviroment for virtual machines.

Q: Why should we use VMs and Docker?

Advantages of VMs

Main Advantages of Software in VMs

Advantages of VMs

Main Advantages of Software in VMs

- Easy to get working software: less maintainance for the developer, less stress for the user

Advantages of VMs

Main Advantages of Software in VMs

- Easy to get working software: less maintainance for the developer, less stress for the user
- Software distributed via VM images works on many OS

Advantages of VMs

Main Advantages of Software in VMs

- Easy to get working software: less maintainance for the developer, less stress for the user
- Software distributed via VM images works on many OS
- Software runs in a predefined enviroment

Advantages of VMs

Main Advantages of Software in VMs

- Easy to get working software: less maintainance for the developer, less stress for the user
- Software distributed via VM images works on many OS
- Software runs in a predefined enviroment
- Easy to keep distributed software up-to-date

Why Docker

Advantages of Docker

Why Docker

Advantages of Docker

- Docker VMs (containers) need few resources

Why Docker

Advantages of Docker

- Docker VMs (containers) need few resources
- Containers can easily interact with the host system (interesting for Jupyter)

Why Docker

Advantages of Docker

- Docker VMs (containers) need few resources
- Containers can easily interact with the host system (interesting for Jupyter)
- Easy distribution of images via Docker Hub

Why Docker

Advantages of Docker

- Docker VMs (containers) need few resources
- Containers can easily interact with the host system (interesting for Jupyter)
- Easy distribution of images via Docker Hub
- Many hosts *(Google, Amazon)* support Docker

Advantages of Docker

Users Point of View

Advantages of Docker

Users Point of View

- Easy to use: start everything with one command

Advantages of Docker

Users Point of View

- Easy to use: start everything with one command
- Almost no performance loss

Advantages of Docker

Users Point of View

- Easy to use: start everything with one command
- Almost no performance loss
- Low disk space requirements

Advantages of Docker

Developers Point of View

Advantages of Docker

Developers Point of View

- Docker is available for many OS, so software can be released in consistent, predefined enviroment and will work for every user

Advantages of Docker

Developers Point of View

- Docker is available for many OS, so software can be released in consistent, predefined enviroment and will work for every user
- Images can be created out of containers, via a `git` flavored interface

Advantages of Docker

Developers Point of View

- Docker is available for many OS, so software can be released in consistent, predefined enviroment and will work for every user
- Images can be created out of containers, via a `git` flavored interface
- Images can be stacked and reused, so the same base image can be used for many applications

Advantages of Docker

Developers Point of View

- Docker is available for many OS, so software can be released in consistent, predefined enviroment and will work for every user
- Images can be created out of containers, via a `git` flavored interface
- Images can be stacked and reused, so the same base image can be used for many applications
  (Example: A basic `GAP` installation with different sets of packages)

Advantages of Docker

Developers Point of View

- Docker is available for many OS, so software can be released in consistent, predefined enviroment and will work for every user
- Images can be created out of containers, via a `git` flavored interface
- Images can be stacked and reused, so the same base image can be used for many applications
  (Example: A basic `GAP` installation with different sets of packages)
- Creation of images can be automated completely via Dockerfiles

Advantages of Docker

Developers Point of View

- Docker is available for many OS, so software can be released in consistent, predefined enviroment and will work for every user
- Images can be created out of containers, via a `git` flavored interface
- Images can be stacked and reused, so the same base image can be used for many applications
  (Example: A basic `GAP` installation with different sets of packages)
- Creation of images can be automated completely via Dockerfiles
- Distribution of images is easy via Docker Hub

Preparation

Install Docker

- Go to `www.docker.com`

Preparation

Install Docker

- Go to `www.docker.com`
- See installation instructions for your OS

Preparation

Install Docker

- Go to `www.docker.com`
- See installation instructions for your OS
- Install Docker

Dockerimage for SPP Computeralgebra

Start a Docker container via

```
docker run -it sppcomputeralgebra/sppdocker
```

Dockerimage for SPP Computeralgebra

Start a Docker container via

```
docker run -it sppcomputeralgebra/sppdocker
```

Software in the Image

Dockerimage for SPP Computeralgebra

Start a Docker container via

```
docker run -it sppcomputeralgebra/sppdocker
```

Software in the Image

- GAP

Dockerimage for SPP Computeralgebra

Start a Docker container via

```
docker run -it sppcomputeralgebra/sppdocker
```

Software in the Image

- GAP (+ undeposited packages)

Dockerimage for SPP Computeralgebra

Start a Docker container via

```
docker run -it sppcomputeralgebra/sppdocker
```

Software in the Image

- GAP (+ undeposited packages)
- Singular

Dockerimage for SPP Computeralgebra

Start a Docker container via

```
docker run -it sppcomputeralgebra/sppdocker
```

Software in the Image

- GAP (+ undeposited packages)
- Singular
- Polymake

Dockerimage for SPP Computeralgebra

Start a Docker container via

```
docker run -it sppcomputeralgebra/sppdocker
```

Software in the Image

- GAP (+ undeposited packages)
- Singular
- Polymake
- Normaliz

Dockerimage for SPP Computeralgebra

Start a Docker container via

```
docker run -it sppcomputeralgebra/sppdocker
```

Software in the Image

- GAP (+ undeposited packages)
- Singular
- Polymake
- Normaliz
- Nemo

Jupyter notebook server

To start a jupyter notebook server, type

```
docker run -t --net="host" \
       sppcomputeralgebra/sppjupyter
```

Jupyter notebook server

To start a jupyter notebook server, type

```
docker run -t --net="host" \
       sppcomputeralgebra/sppjupyter
```

Once the server is running, open your browser on page

```
localhost:8888
```

Jupyter notebook server

To start a jupyter notebook server, type

```
docker run -t --net="host" \
       sppcomputeralgebra/sppjupyter
```

Once the server is running, open your browser on page

```
localhost:8888
```

Software with (experimental) Jupyter kernels

- GAP (stable, by Markus Pfeiffer)

Jupyter notebook server

To start a jupyter notebook server, type

```
docker run -t --net="host" \
       sppcomputeralgebra/sppjupyter
```

Once the server is running, open your browser on page

```
localhost:8888
```

Software with (experimental) Jupyter kernels

- GAP (stable, by Markus Pfeiffer)
- Singular (alpha)

Jupyter notebook server

To start a jupyter notebook server, type

```
docker run -t --net="host" \
       sppcomputeralgebra/sppjupyter
```

Once the server is running, open your browser on page

```
localhost:8888
```

Software with (experimental) Jupyter kernels

- GAP (stable, by Markus Pfeiffer)
- Singular (alpha)
- Polymake (alpha)

Jupyter notebook server

To start a jupyter notebook server, type

```
docker run -t --net="host" \
       sppcomputeralgebra/sppjupyter
```

Once the server is running, open your browser on page

```
localhost:8888
```

Software with (experimental) Jupyter kernels

- GAP (stable, by Markus Pfeiffer)
- Singular (alpha)
- Polymake (alpha)
- Soon: Julia/Nemo (by Julia community)

Basic Steps for Distributing

- Select a base image, like the spp image

```
docker pull sppcomputeralgebra/sppdocker
```

Basic Steps for Distributing

- Select a base image, like the spp image

```
docker pull sppcomputeralgebra/sppdocker
```

- Start a container from that image

```
docker run --name="my_container" \
           -it sppcomputeralgebra/sppdocker
```

Basic Steps for Distributing

- Select a base image, like the spp image

```
    docker pull sppcomputeralgebra/sppdocker
```

- Start a container from that image

```
    docker run --name="my_container" \
               -it sppcomputeralgebra/sppdocker
```

- Install additional software

Basic Steps for Distributing

- Select a base image, like the spp image

```
docker pull sppcomputeralgebra/sppdocker
```

- Start a container from that image

```
docker run --name="my_container" \
           -it sppcomputeralgebra/sppdocker
```

- Install additional software
- Create image out of the container, containing the installed software

```
docker commit my_container sebasguts/my_image
```

Basic Steps for Distributing

- Select a base image, like the spp image

```
docker pull sppcomputeralgebra/sppdocker
```

- Start a container from that image

```
docker run --name="my_container" \
           -it sppcomputeralgebra/sppdocker
```

- Install additional software
- Create image out of the container, containing the installed software

```
docker commit my_container sebasguts/my_image
```

- Release it by uploading it to Docker Hub

```
docker push sebasguts/my_image
```

Basic Steps for Distributing

- Select a base image, like the spp image

```
docker pull sppcomputeralgebra/sppdocker
```

- Start a container from that image

```
docker run --name="my_container" \
           -it sppcomputeralgebra/sppdocker
```

- Install additional software
- Create image out of the container, containing the installed software

```
docker commit my_container sebasguts/my_image
```

- Release it by uploading it to Docker Hub

```
docker push sebasguts/my_image
```