Welcome!

Poll: Who knows Virtualization/Sandboxing/Docker?

Presentation is ess technical than usual.

Although might be a bit linux centered.

Presentation is done in jovial tone.

Definitions

Deployment

`Def:` The process of distributing software to client machines.

Sandboxing

`Def:` Environment for application that provide limited resource access.

Before we actually start, there are some terms that need clarification.

How Deployment was done around 1950

So, what happens if we're a company that has written some server side software, which needs to be installed on the client's hardware? Our software has many dependencies, might rely on certain software versions of, for example, Apache.

`Certain dependencies might still have bugs.`

Software deployment meant...

- ...to install the software on the clients hardware.
- ...to use the software's buildsystem.
- ...and to install it's dependencies.
- ...using the dependencie's buildsystems.
- ...fixing possible bugs on the client's platform.
- ...fixing possible dependencies bugs.
- ...paying treatment cost developer's burnout.

A bit more recent

- Hey, lets use virtualization for testing!
- Simulate the clients environment in a virtual machine.
- Test it in the VM, just copy the tested software over.
- What if the clients environment changes?

Try again

- Hey, just lets ship the Virtual Machine Image!
- Works.
- But feels like an awful hack.
- Deploying a 5GB+ VM image on every software update.
- Seriously?

Docker to the rescue

- Containers instead of Images.
- Deploying the `diff` instead of the whole container.
- Base images for many popular linux distributions.
- New base images can be uploaded to DockerHub.
- Docker is the application engine that is able to run containers.

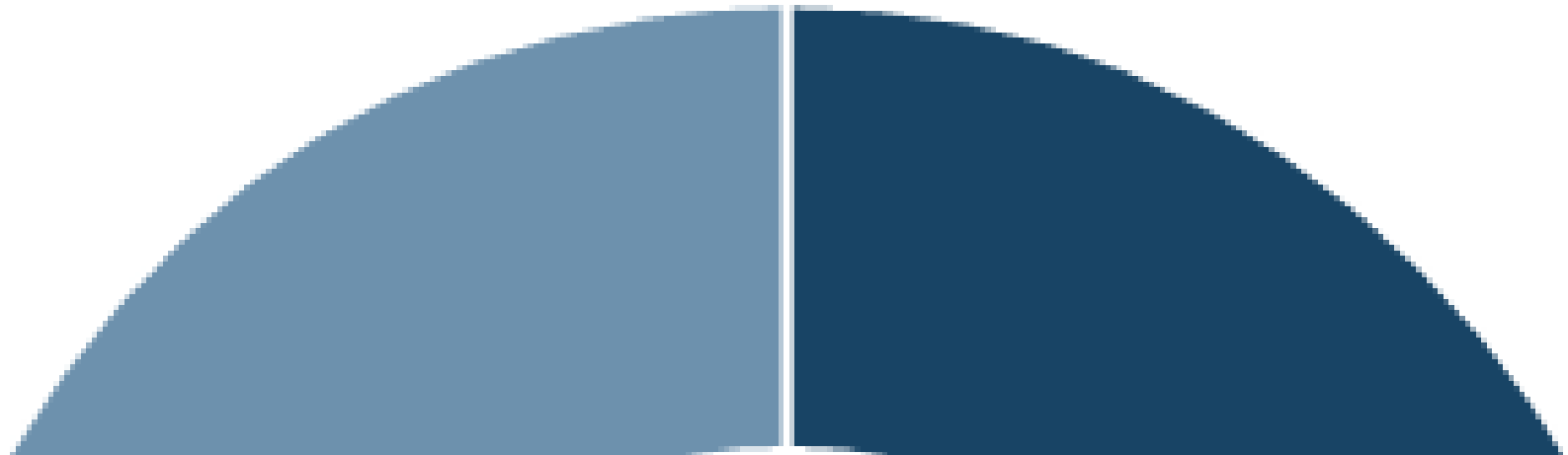
Features

- Very fast booting containers (same Kernel!)
- Containerized applications run sandboxed.
- Easy to pack own applications in containers.
- Support for creating own Images.
- Built-in versioning support.
- Divided in Daemon and Docker clients.
- Linking containers together.
- ...

shipyard

Dashboard Containers

CPU



Technical Stuff

- Focus on processes, not on virtualizing operating systems.
- One process per container.

Docker builds on Linux features:

- **cgroups:** Grouping processes together.
- **namespaces:** Separate processes in own namespaces.
- **LXC:** Combines both to provide *Operating system-level virtualization.*
- **aufs:** Overlay file system.

Demonstration

```
$ docker images
$ docker run base/arch echo hello augsburg!
$ uname -a
$ docker run -t -i base/arch /bin/bash
 uname -a
 rm -r /bin
$ docker diff
$ docker hub
```

Usecases

- Deployment.
- Sandboxing applications.
- Testbed for application developement.
  Personal usecase: running a self-written duplicate finder on /
  System still working? No? Just restart the container.
- Cluster Management with `CoreOS`.
- ...

So, docker is the new thing?

Depends on your usecase.

- Still can only run Linux based containers.
- But it can run them on Windows/MacOSX using lightweight virtualization. (`boot2docker`).
- Microsoft has plans to port Docker fully to Windows.
- And even to use Windows as Image.
- What about GUI applications?

The real valu

Are there areas where virtualization is still needed?

Yes.

References

. . .

Thank you for your attention

Hooray, school's out!