eurac research

Containerization and why you should care

A short introduction to containerization

Alexander König, Egon Stemle
<{alexander.koenig, egon.stemle}@eurac.edu>

Outline

Deploying applications

Criteria

- Fast

Deploying applications

Criteria

- Fast
- Reproducable

Deploying applications

Criteria

- Fast
- Reproducable
- Easy to (re-)deploy (e.g. after security fixes)

Deploying applications

Criteria

- Fast
- Reproducable
- Easy to (re-)deploy (e.g. after security fixes)
- (Easily) scalable

Outline

”Bare Metal”

The Past

Characteristics of bare metal deployments

- Application is deployed on a real computer

”Bare Metal”

The Past

Characteristics of bare metal deployments

- Application is deployed on a real computer
- If more than one app is deployed on the same computer you might get interferences

”Bare Metal”

The Past

Characteristics of bare metal deployments

- Application is deployed on a real computer
- If more than one app is deployed on the same computer you might get interferences
  - Different versions of the same library might be needed

”Bare Metal”

The Past

Characteristics of bare metal deployments

- Application is deployed on a real computer
- If more than one app is deployed on the same computer you might get interferences
  - Different versions of the same library might be needed
  - One app crashing might drag another one down with it

”Bare Metal”

The Past

Characteristics of bare metal deployments

- Application is deployed on a real computer
- If more than one app is deployed on the same computer you might get interferences
  - Different versions of the same library might be needed
  - One app crashing might drag another one down with it
- If only one app is deployed on a computer you might waste a lot of resources

Characteristics of deployments on a virtual machine

- Multiple virtual computers deployed on one actual computer

Virtualization

The Present

Characteristics of deployments on a virtual machine

- Multiple virtual computers deployed on one actual computer
- Each application is in its own virtual environment (= no interference)

Characteristics of deployments on a virtual machine

- Multiple virtual computers deployed on one actual computer
- Each application is in its own virtual environment (= no interference)
- Easy to start applications, also multiple versions of the same one (for scalability)

Characteristics of deployments on a virtual machine

- Multiple virtual computers deployed on one actual computer
- Each application is in its own virtual environment (= no interference)
- Easy to start applications, also multiple versions of the same one (for scalability)
- Severe implications to performance because of the extra layer

Characteristics of deployments on a virtual machine

- Multiple virtual computers deployed on one actual computer
- Each application is in its own virtual environment (= no interference)
- Easy to start applications, also multiple versions of the same one (for scalability)
- Severe implications to performance because of the extra layer
- Example software: VMWare, Virtualbox, etc.

Characteristics of containerized deployments

- Apps are deployed in a so-called container

Characteristics of containerized deployments

- Apps are deployed in a so-called container
- Imitates a virtual computer (app is sandboxed away from all other apps)

Characteristics of containerized deployments

- Apps are deployed in a so-called container
- Imitates a virtual computer (app is sandboxed away from all other apps)
- But is much more lightweight than traditional Virtual Machines

Characteristics of containerized deployments

- Apps are deployed in a so-called container
- Imitates a virtual computer (app is sandboxed away from all other apps)
- But is much more lightweight than traditional Virtual Machines

- Docker, Docker Swarm, Kubernetes

Outline

Docker

Docker features

- A way to deploy an application in a sandboxed virtual environment

Docker

Docker features

- A way to deploy an application in a sandboxed virtual environment
- Uses the host system's kernel to access the actual hardware

Docker

Docker features

- A way to deploy an application in a sandboxed virtual environment
- Uses the host system's kernel to access the actual hardware
- Often uses stripped-down environments to reduce system load

Docker

Docker features

- A way to deploy an application in a sandboxed virtual environment
- Uses the host system's kernel to access the actual hardware
- Often uses stripped-down environments to reduce system load
- A specific setup (OS, libraries, helper applications) is built according to a recipe (Dockerfile) and stored as an image

Docker

Docker features

- A way to deploy an application in a sandboxed virtual environment
- Uses the host system's kernel to access the actual hardware
- Often uses stripped-down environments to reduce system load
- A specific setup (OS, libraries, helper applications) is built according to a recipe (Dockerfile) and stored as an image
- Images can be uploaded to a registry on the web (e.g. hub.docker.com or your own gitlab instance) and be fetched from there

Docker

Docker features

- A way to deploy an application in a sandboxed virtual environment
- Uses the host system's kernel to access the actual hardware
- Often uses stripped-down environments to reduce system load
- A specific setup (OS, libraries, helper applications) is built according to a recipe (Dockerfile) and stored as an image
- Images can be uploaded to a registry on the web (e.g. hub.docker.com or your own gitlab instance) and be fetched from there
- Philosophy of "one process per container", which means that logically distinct sub processes (e.g. a database) should live in their own separate Container

Docker Features I

Security

- Only explicitly configured ports will be exposed to the outside world

Docker Features I

Security

- Only explicitly configured ports will be exposed to the outside world
- You have full control over what services run in your container, often even standard services like cron or systemd are not installed by default

Docker Features II

Reproducibility

- The exact setup is documented in a Dockerfile (basically a recipe for how to create this virtual environment)

Docker Features II

Reproducibility

- The exact setup is documented in a Dockerfile (basically a recipe for how to create this virtual environment)
- Dockerfile guarantees that you can easily recreate the same setup

Docker Features II

Reproducibility

- The exact setup is documented in a Dockerfile (basically a recipe for how to create this virtual environment)
- Dockerfile guarantees that you can easily recreate the same setup
- As a bonus the Dockerfile is human readable, so it's easy to understand what is happening

Docker Features III

Scalability

- You can easily create as many containers as you need from the same Docker image

Docker Features III

Scalability

- You can easily create as many containers as you need from the same Docker image
- There are deployment systems that automate this (e.g. Kubernetes)

Docker Features III

Scalability

- You can easily create as many containers as you need from the same Docker image
- There are deployment systems that automate this (e.g. Kubernetes)
- A lot of web hosters allow you to deploy Docker images on their cloud (AWS, Google Cloud, Microsoft Azure,...)

Kubernetes

Kubernetes features

- Can orchestrate complicated setup with multiple interdependent containers

Kubernetes

Kubernetes features

- Can orchestrate complicated setup with multiple interdependent containers
- Runs on a cluster of servers that is easily extended

Kubernetes

Kubernetes features

- Can orchestrate complicated setup with multiple interdependent containers
- Runs on a cluster of servers that is easily extended
- Can easily scale up and down by automatically starting more instances of a container if needed

Docker Example

Dockerized Pybossa

```
FROM redis:3.2

# install git & postgres
RUN apt-get update && apt-get -y upgrade && apt-get install -y git libpq-dev python-psycopg2 libsasl2-dev libldap2-dev libssl-dev

# install python stuff
RUN apt-get update && apt-get install -y python-dev build-essential libjpeg-dev libssl-dev libffi-dev python-pip dbus libdbus-1-dev libdbus-glib-1
-dev libldap2-dev libsasl2-dev

# force check for updates in git repo
ADD https://api.github.com/repos/Scifabric/pybossa/git/refs/heads/master version_ctap.json
# clone git repo
RUN git clone --recursive https://github.com/Scifabric/pybossa /opt/pybossa
# Access the source code folder
WORKDIR /opt/pybossa
# Upgrade pip to latest version
RUN pip install -U pip
# Install the required libraries
RUN pip install -r requirements.txt

# install redis
RUN apt-get install -y redis-server

# install supervisor
RUN apt-get install -y python-setuptools
RUN easy_install supervisor
COPY supervisord.conf /etc/
RUN mkdir -p /var/log/supervisor

# copy over config files
COPY settings_local.py /opt/pybossa/
COPY alembic.ini /opt/pybossa/
COPY sentinel.conf /opt/pybossa/contrib/

# run entrypoint script
COPY entrypoint.sh /usr/bin/entrypoint.sh
RUN chmod u+x /usr/bin/entrypoint.sh
ENTRYPOINT ["/usr/bin/entrypoint.sh"]
                                                                    39,0-1        All
```

See `https://gitlab.inf.unibz.it/commul/docker/pybossa`