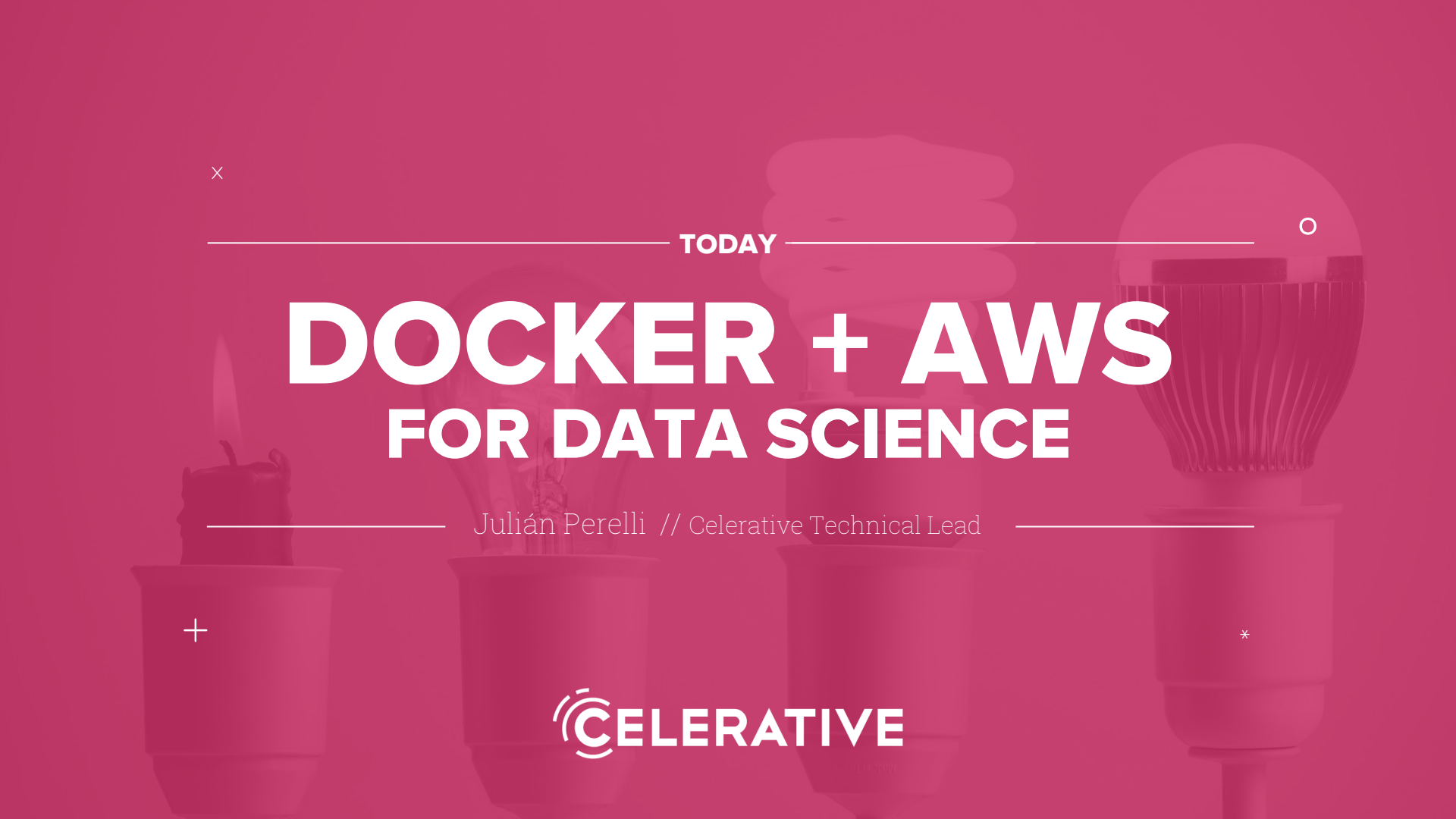

TODAY
DOCKER + AWS
FOR DATA SCIENCE
Julián Perelli // Celerative Technical Lead
CELERATIVE

# WHAT'S THIS TALK ABOUT?

Ease complex software installation

- Server or libraries/dependencies configuration
- Backup and restore
- Deploy fast, developer faster
- **Avoid wasting time** in infrastructure problems

# SUMMARY

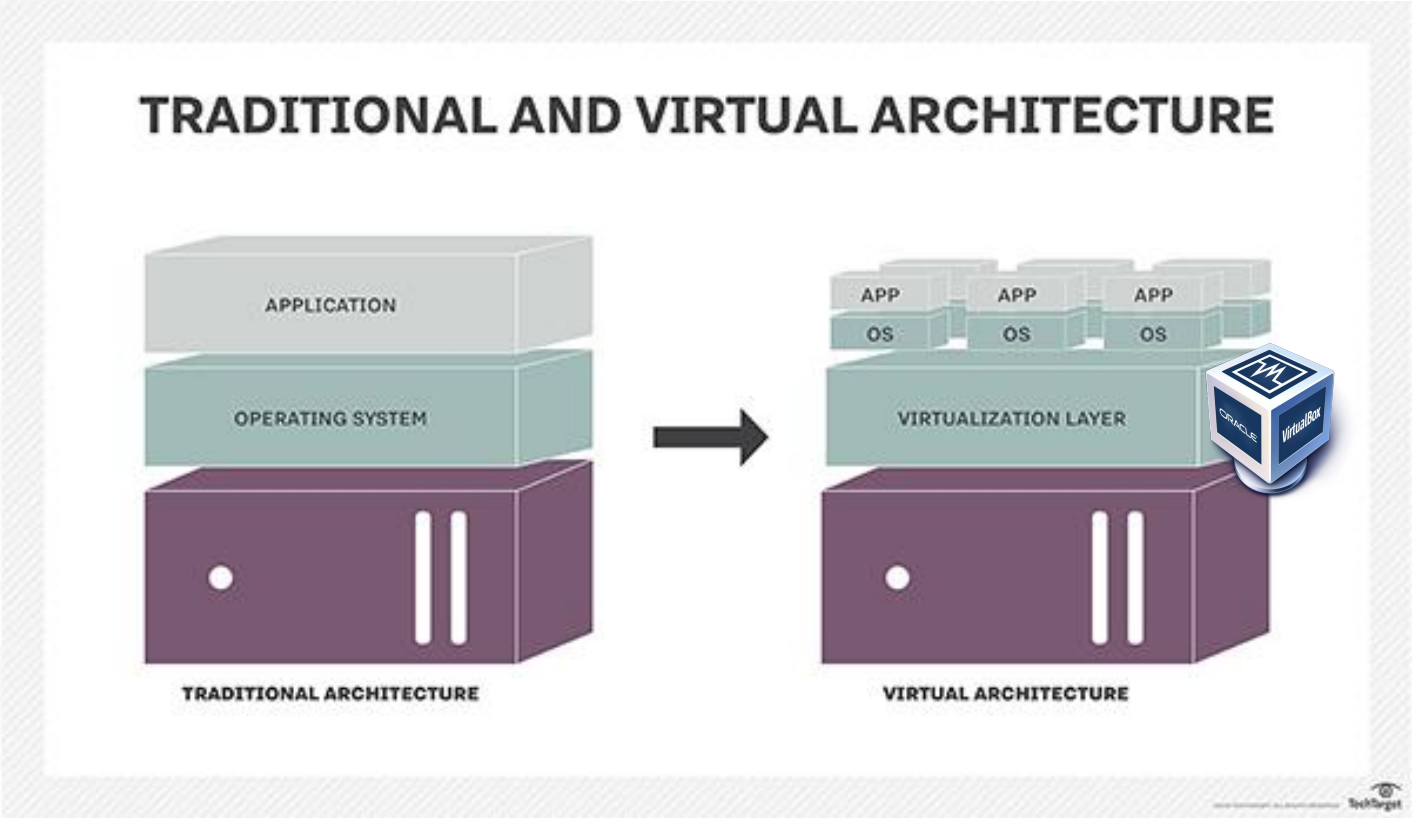
- Virtual machine concept
- Docker
- Docker deploy in AWS ECR+ECS
- Docker compose
- Docker compose deploy in AWS EC2

# VIRTUAL MACHINE

# VIRTUAL MACHINE

Emulation of a computer
inside your computer

# TRADITIONAL VS VIRTUALIZED



# VIRTUALIZED APPS

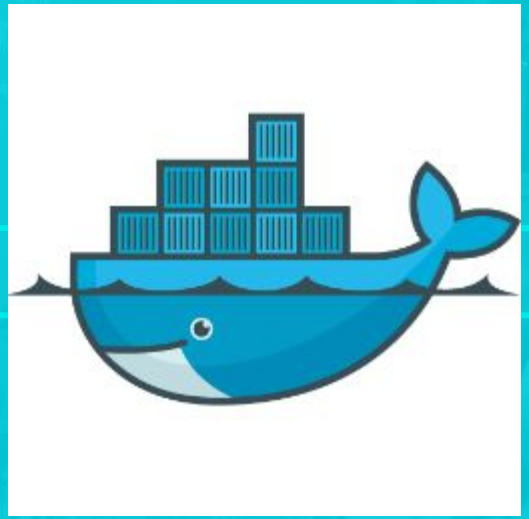
Sandbox OS and software per app

- Manage software dependencies
- Backup and restore
- Share pre-installed OS
- Copies / run multiple instances

Automatable

- Installation
- Running

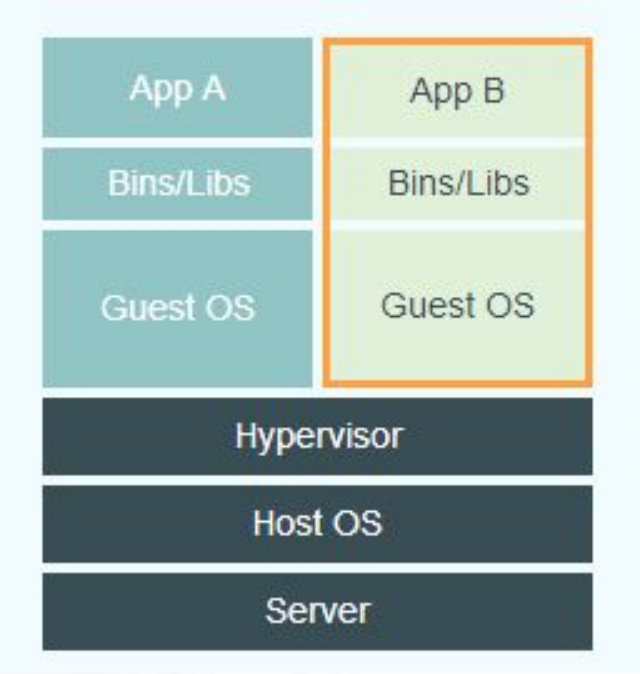
# DOCKER

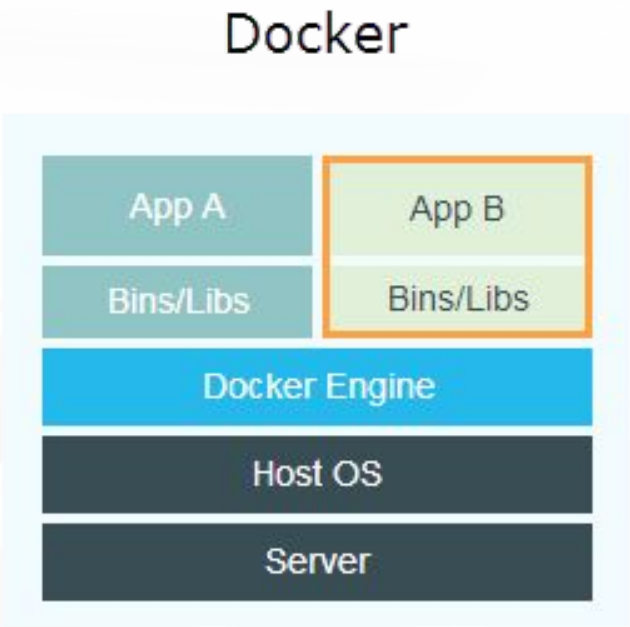


DOCKER

# Containerized applications

# VIRTUAL VS DOCKER



# WHAT DOES DOCKER PROVIDE?

- Zero startup time
- Sandboxing
  - Backup and restore
  - Run hazardous apps without fear
  - Start and run multiple copies
- Share preinstalled/configured “images”
- Repositories FTW!

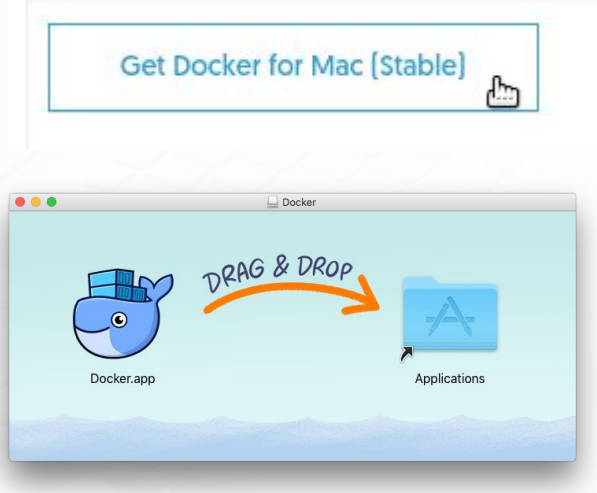
  https://hub.docker.com/explore/

# DOCKER TUTORIAL

# INSTALL DOCKER

**Linux:** apt install docker

**OSx:** https://docs.docker.com/docker-for-mac/install/




# FLASK HELLO DOCKER

**flaskapp/app/main.py**

```python
from flask import Flask
app = Flask(__name__)

@app.route("/")
def hello():
    return "Hello World from Flask"

if __name__ == "__main__":
    # Only for debugging while developing
    app.run(host='0.0.0.0', debug=True, port=80)
```

$ **docker run -p 80:80 tiangolo/uwsgi-nginx-flask:python3.6**

# DOCKER CUSTOM TUTORIAL

# FLASK HELLO DOCKER CUSTOM

**flaskapp/Dockerfile**

```
FROM tiangolo/uwsgi-nginx-flask:python3.6
RUN pip install pymongo
COPY ./app /app
```

$ **docker build . -t myflaskimage**

**flaskapp/app/main.py**

```
from flask import Flask
import pymongo
app = Flask(__name__)

@app.route("/")
def hello():
    return "Hello pymongo! Version {}".format(pymongo.__version__)

if __name__ == "__main__":
    # Only for debugging while developing
    app.run(host='0.0.0.0', debug=True, port=80)
```

$ **docker run -p 80:80 myflaskimage**

# DOCKER CUSTOM DEPLOY AWS

# DOCKER DEPLOY TO AWS

[ECR] EC2 Container Repository

[ECS] EC2 Container Service

# UPLOAD TO ECR (REPOSITORY)

**docker build**
docker build . -t myflaskimage

**docker tag**
docker tag myflaskimage 532041945183.dkr.ecr.us-west-2.amazonaws.com/**myflaskimage**:latest

**aws login**
aws ecr get-login --region us-west-2
(download aws cli from http://docs.aws.amazon.com/cli/latest/userguide/installing.html)

**docker upload to aws ECR (repository)**
docker push 532041945183.dkr.ecr.us-west-2.amazonaws.com/myflaskimage:latest

# RUN CONTAINER IN ECS

**ecs-deploy.sh**
https://github.com/silinternational/ecs-deploy

```
./ecs-deploy.sh
    -c my-cluster
    -n myflaskimage
    -i 532041945183.dkr.ecr.us-west-2.amazonaws.com/myflaskimage:latest
```

# DOCKER COMPOSE

# DOCKER COMPOSE

A tool for defining and running multi-container Docker applications

# FLASK HELLO DOCKER-COMPOSE

**flask-sample/docker-compose.yml**

```yaml
version: '2'
services:

  flask:
    build: flaskapp
    ports:
      - 8080:80
    links:
      - database

  database:
    image: mongo
    volumes:
      - ./volume-mongodb:/data/db
```

**flask-sample/flaskapp/app/main.py**

```python
from flask import Flask
from pymongo import MongoClient
app = Flask(__name__)

@app.route("/")
def hello():
    mongo = MongoClient('database')
    dbnames = mongo.database_names()
    return "Hello pymongo! dbnames: {}".format(dbnames)

if __name__ == "__main__":
    # Only for debugging while developing
    app.run(host='0.0.0.0', debug=True, port=80)
```

$ **docker-compose up**

# DOCKER COMPOSE DEPLOY AWS

# UPLOAD TO EC2

0. **https://stackoverflow.com/questions/6394762/how-to-setup-ssh-access-for-amazon-ec2-instance**
1. cd flask-sample
2. scp -r . root@ipaddr:/home/root/
3. ssh root@ipaddr
4. docker-compose up --build -d

# THANK YOU!

julian.perelli@celerative.com

jperelli

44 Tehama St. / San Francisco / CA

us@celerative.com

celerative.com <<

42 nº 1389 / La Plata / Arg.

info@celerative.com

+54 (011) 527 56155